EPFL

Plan d’études

SYSTEMES DE COMMUNICATION

2024-2025

arrêté par la vice-présidence académique de l'EPFL le 20 juin 2024

Directeur de section	**Prof. K. Aberer**
Adjointe du directeur de section	**Mme E. Hazboun**
Responsable passerelle HES	**Mme E. Hazboun**
Délégué à la mobilité	**M. G. Regev**
Coordinatrice des stages en entreprise	**Mme P. Genet**
Spécialiste administrative pour le Bachelor	**Mme M. Emery**
Spécialiste administrative pour le Master	**Mme C. Dauphin**

Aux cycles bachelor et master, selon les besoins pédagogiques, les heures d’exercices mentionnées dans le plan d’études pourront être intégrées dans les heures de cours ; les scolarités indiquées représentent les nombres moyens d’heures de cours et d’exercices hebdomadaires sur le semestre.

2024-2025

SYSTÈMES DE COMMUNICATION

Cycle Propédeutique

Cursus commun IN-SC

Code	Matières	Type de branches	Enseignant·es sous réserve de modification	Semestres BA1 c	BA1 e	BA1 p	BA2 c	BA2 e	BA2 p	Coeff.	Période des épreuves	Type examen
Bloc 1										**38**		
	Semestre automne									25		
CS-101	Advanced information, computation, communication I	Spécifique	Bourgeat/Käser	**4**	**2**					7	H	écrit
MATH-111(a)	Algèbre linéaire (en français) ou	Polytechnique	Basterrechea	**4**	**2**					6	H	écrit
MATH-111(en)	Algèbre linéaire (en anglais)		Iseli									
MATH-111(pi)	Algèbre linéaire (classe inversée)		Deparis									
MATH-101(a)	Analyse I (en français) ou	Polytechnique	Mila	**4**	**2**					6	H	écrit
MATH-101(de)	Analyse I (en allemand) ou		Krieger									
MATH-101(en)	Analyse I (en anglais)		Mountford									
MATH-101(pi)	Analyse I (classe inversée)		Friedli									
PHYS-101(h)	Physique générale : mécanique (en français) ou	Polytechnique	Hogge	**4**	**2**					6	H	écrit
PHYS-101(de)	Physique générale : mécanique (en allemand) ou		Gruetter									
PHYS-101(en)	Physique générale : mécanique (en anglais) ou		Baquero Ruiz									
PHYS-101(pi)	Physique générale : mécanique (classe inversée)		Hébert									
	Semestre printemps											
COM-102	Advanced information, computation, communication II	Spécifique	Gastpar				**4**	**2**		7	E	écrit
MATH-106(e)	Analyse II (en français) ou	Polytechnique	Lachowska				**4**	**2**		6	E	écrit
MATH-106(en)	Analyse II (en anglais)		Richter									
Bloc 2										**23**		
	Semestre automne											
CS-107	Introduction à la programmation	Polytechnique	Pereira Pires/Sam	**2**	**2**	**1**				5	sem A	
	Semestre printemps											
CS-173	Fundamentals of digital systems	Spécifique	Stojilovic				**3**	**4**		7	E	écrit
ENV-101	Durabilité	Polytechnique	Chapellaz + Divers enseignants				**2**			2	sem P	
CS-108	Pratique de la programmation orientée-objet	Spécifique	Schinz				**2**	**2**	**6**	9	sem P	
	Totaux :			**18**	**10**	**1**	**15**	**10**	**6**	**61**		
	Totaux par semaine :				**29**			**31**				

Remarques :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

Les cours en allemand et en anglais sont disponibles sous réserve de la compatiblité des horaire des cours

2024-2025 # SYSTÈMES DE COMMUNICATION **Cycle Bachelor**

Code	Matières	Enseignant·es sous réserve de modification	Semestres BA3 c	BA3 e	BA3 p	BA4 c	BA4 e	BA4 p	BA5 c	BA5 e	BA5 p	BA6 c	BA6 e	BA6 p	Crédits 2e	Crédits 3e	Nbre places	Période des épreuves	Type examen
Bloc A															**22**				
CS-250	Algorithms I	Chiesa/Svensson				**4**	**2**								8			E	écrit
MATH-203(d)	Analysis III	Licht	**3**	**2**											6			H	écrit
COM-202	Signal processing	Prandoni/Shkel				**4**	**2**	**2**							8			E	écrit
Bloc B															**24**				
CS-202	Computer systems	Argyraki/Chappelier/ Bugnion				**4**	**2**	**2**							8			E	écrit
PHYS-114	Physique générale : électromagnétisme	Avino	**2**	**2**											4			H	écrit
CS-233	Introduction to machine learning	Fua/Salzmann				**2**	**2**	**2**							6			E	écrit
MATH-232	Probability and statistics (for IC)	Abbé	**4**	**2**											6			H	écrit
Bloc C																**22**			
MATH-310	Algebra	Lachowska							**2**	**2**						4		H	écrit
COM-301	Computer security and privacy	Bugnion/Troncoso							**2**	**1**	**1**					6		H	écrit
COM-300	Modèles stochastiques pour les communications	Thiran							**4**	**2**						6		H	écrit
COM-302	Principles of digital communications	Telatar										**4**	**1**	**1**		6		E	écrit
Groupe "Projet" et "options"															**44**				
- Groupe "Projet" (choisir l'une des deux propositions)																**8**			
COM-304	Communications Project	Al Hassanieh/Zamir										**2**		**10**		8		sem P	sans retrait
COM-307	Projet de recherche en Systèmes de Communication I	Divers enseignants							←			2		→		8		sem A ou P	
- Groupe options			←						36					→	**10**	**26**			
MATH-207(d)	Analysis IV	Licht				2	2								4			E	écrit
CS-200	Computer architecture	Ienne	4	2	2										8			H	écrit
EE-202b	Electronique I	Zysman/Sallèse	2	1											4			sem A	
CS-213	Interaction personne-système	Dillenbourg				2		2							5			E	écrit
BIO-109	Introduction aux sciences du vivant (pour IC)	Zufferey R.				4	2								6			E	écrit
CS-290	Responsible software	Hardebolle	2	2					2	2					4	4		sem A	
PHYS-202	Mécanique analytique	Jacquod	3	2											5			H	écrit
CS-328	Numerical methods for visual computing and ML (pas donné en 24-2025)	Jakob	2	2											4			H	écrit
CS-214	Software construction	Kuncak/Odersky/ Pit-Claudel	3	2	3										8			H	écrit
CS-234	Technologies for democratic society	Ford	2	1	2										5			H	écrit
CS-341	Computer graphics	Pauly				2	1	2							6			E	écrit
CS-320	Computer language processing	Kuncak										2	2	2		6		sem P	
CS-300	Data-Intensive systems	Ailamaki/Kashyap										2	1	2		6		E	écrit
EE-203b	Electronique II (pour IC)	Zysman/Sallèse							2	2						4		sem A	
EE-200	Electromagnétisme I : lignes et ondes	Fleury							2	1						3		H	écrit
EE-201	Electromagnétisme II : calcul des champs	Fleury										2	1			3		E	écrit
CS-330	Intelligence artificielle	Faltings										2	2			4		sem P	
COM-308	Internet analytics (pas donné en 24-25)	Grossglauser										2	2			6		E	écrit
CS-308	Introduction to quantum computation	Lévêque/Urbanke										3	1			5		E	écrit
COM-309	Introduction to quantum information processing	Macris							3	1						5		H	écrit
PHYS-344	Quantum mechanics for non-physicists	Manucharyan							2	2						5		H	oral
CS-358(a)	Making Intelligent Things A	Koch								1	12					8	50	sem A	sans retrait
CS-358(b)	Making Intelligent Things B	Koch											1	12		8	75	sem P	sans retrait
BIOENG-310	Neuroscience foundations of engineers	Schrimpf/Zenk										3	3			6		E	écrit
CS-302	Parallelism & concurrency in Software	Falsafi										3	3			6		sem P	
EE-310	Systèmes embarqués microprogrammés	Atienza							2		2					4	80	H	oral sans retrait
CS-311	The Software enterprise - from ideas to products	Candea/Bugnion							2	1	10					8		sem A	
CS-251	Theory of computation	Göös										2	2			6		E	écrit
"SHS et MGT transversal" :															**8**				
HUM/MGT-nnn	SHS : Cours à choix I selon Plan d'études SHS & MGT	Divers enseignants	2												2			sem A	
HUM/MGT-nnn	SHS : Cours à choix II selon Plan d'études SHS & MGT	Divers enseignants				2									2			sem P	
HUM/MGT-nnn	SHS : Cours à choix III selon Plan d'études SHS & MGT	Divers enseignants							2							2		sem A	
HUM/MGT-nnn	SHS : Cours à choix IV selon Plan d'études SHS & MGT	Divers enseignants										2				2		sem P	
	Totaux:														**60**	**60**			

Remarques :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** sans retrait = pas de retrait possible après le délai d'inscription

2024-2025 SYSTÈMES DE COMMUNICATION **Cycle Master**

Code	Matières	Enseignant·es sous réserve de modification	Spécialisations***										Semestres MA1/MA3			Semestres MA2/MA4			Crédits	Période des épreuves*	Type examen**
			A	B	C	D	E	F	G	H	I	J	c	e	p	c	e	p			
Groupe "Core courses et options"																			**72**		
Groupe 1 "Core courses"																			**min. 32**		
COM-417	Advanced probability and applications	Shkel		B						H	I		**4**	**2**					8	H	écrit
CS-450	Algorithms II	Kapralov/Svensson		B	C	D	E				I		**4**	**3**					8	H	écrit
COM-401	Cryptography and security	Vaudenay				D	E					J	**4**	**2**					8	H	écrit
CS-451	Distributed algorithms	Guerraoui			C		E		G		I	J	**2**	**1**	**3**				8	H	écrit
COM-402	Information security and privacy	Payer		B		D	E		G				**3**	**1**	**2**				8	H	écrit
COM-404	Information theory and coding	Telatar		B						H	I		**4**	**2**					8	H	écrit
CS-433	Machine learning	Jaggi/Flammarion		B				F			I	J	**4**	**2**	**2**				8	H	écrit
COM-405	Mobile networks	Al Hassanieh				D	E		G	H			**3**	**2**	**2**				8	H	écrit
COM-430	Modern digital communications: a hands-on approach	Chiurtu					E	F		H			**2**		**2**				8	sem A	
COM-500	Statistical signal and data processing through applications	Ridolfi		B				F		H						**3**	**2**		8	E	écrit
COM-407	TCP/IP networking	Nikolopoulos				D	E		G	H			**2**	**2**	**2**				8	H	écrit
Groupe 2 "Options" (la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)																					
	- voir liste cours à optinos																				
Bloc "Projets et SHS" :																			**18**		
COM-416	Research project in communication systems II	divers enseignants											←		**2**			→	12	sem A ou P	
HUM-nnn	SHS : introduction au projet	divers enseignants											**2**		**1**				3	sem A	
HUM-nnn	SHS : projet	divers enseignants																**3**	3	sem P	sans retrait
Total des crédits du cycle master																			**90**		

Remarques :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** sans retrait = pas de retrait possible après le délai d'inscription

*****Spécialisations :**

A : Computer Engineering
B : Data Analytics
C : Foundations of Software
D : Cyber security - SP
E : Networking and Mobility
F : Signals, Images, and Interfaces
G : Software Systems
H : Wireless Communications
I : Computer Science Theory
J : Internet Information Systems

Stage d'ingénieur :

Voir les modalités dans le règlement d'application

Mineurs :

Le cursus peut être complété par un des mineurs figurant dans l'offre de l'EPFL (renseignements à la page sac.epfl.ch/mineurs),
à l'exclusion des mineurs, "Data Science", "Informatique", "Cyber security" et "Systèmes de communication" qui ne peuvent pas être choisi.
Le choix des cours de tous les mineurs se fait sur conseil de la section de l'étudiant et du responsable du mineur.

2024-2025 **SYSTÈMES DE COMMUNICATION - Options**

Code	Matières	Enseignant·es sous réserve de modification	Spécialisations***										Semestres MA1/MA3			MA2/MA4			Crédits	Nbre place	Période des épreuves*	Type examen**
			A	B	C	D	E	F	G	H	I	J	c	e	p	c	e	p				
Groupe 2 "Options" (la somme des crédits des groupes 1 et 2 doit être de 72 cr																						
CS-500	AI product management	Kaboli/Zamir											*2*	*2*	*3*				6		sem A	sans retrait
CS-420	Advanced compiler construction	Schinz	A		C				G							*2*		*2*	6		sem P	
CS-470	Advanced computer architecture	Ienne	A			D			G							3		2	8		E	écrit
CS-440	Advanced computer graphics (pas donné en 24-25)	Jakob						F								2	2		6		sem P	
COM-501	Advanced cryptography	Vaudenay				D										2	2		6		E	écrit
CS-471	Advanced multiprocessor architecture	Falsafi	A						G				4		8				8		sem A	
CS-477	Advanced operating systems	Kashyap	A		C	D			G				2	1	2				6		H	écrit
CS-523	Advanced topics on privacy enhancing technologies	Troncoso				D										3	1	2	8		E	écrit
EE-512	Applied biomedical signal processing	Lemay						F					2		2				4		H	écrit
MATH-493	Applied biostatistics	Goldstein														2	2		5		sem P	
CS-401	Applied data analysis	Brbic		B									2		2				8		H	écrit
EE-554	Automatic speech processing	Magimai Doss						F					2	2					4		H	écrit
MICRO-452	Basics of mobile robotics	Mondada											1		3				4		H	écrit
BIO-410	Bioimage informatics	Sage/Seitz						F								2		2	4		E	écrit
MGT-416	Causal inference ****	Kiyavash														2	1		4		sem P	
MATH-352	Causal thinking	Stensrud											2	2					5		H	écrit
BIO-105	Cellular biology and biochemistry for engineers	Zufferey											2	2					4		H	écrit
CS-524	Computational complexity	Göös		B	C						I		3	1					6		H	écrit
NX-465	Computational neuroscience : neural dynamics	Gerstner														2	2		5		E	écrit
CS-413	Computational photography	Süsstrunk						F								2		2	6		sem P	
COM-418	Computers and Music (pas donné en 24-25)	Prandoni						F								2	1		6		sem P	
CS-442	Computer vision	Fua						F								2	1		6		E	écrit
CS-453	Concurrent computing	Guerraoui			C				G		I		2	1	2				6		H	écrit
COM-480	Data visualization	Vuillon		B				F								2		2	6		sem P	
CS-438	Decentralized systems engineering	Borso/Ford							G				2	2	2				8		H	écrit
EE-559	Deep learning	Caballaro						F								2	2		4	150	sem P	sans retrait
CS-456	Deep reinforcement learning	Gulcehre														2	1	1	6		E	écrit
CS-472	Design technologies for integrated systems	De Micheli	A										3		2				6		sem A	
CS-411	Digital education	Dillenbourg/Jermann						F								2		2	6		E	écrit
CS-423	Distributed information systems	Aberer		B			E					J	2	1	1				6		H	écrit
ENG-466	Distributed intelligent systems	Martinoli	A										2	3					5	40	H	oral sans retrait
COM-502	Dynamical system theory for engineers ****	Thiran P.														3	1		6		E	écrit
CS-476	Embedded systems design	Kluter	A													2		2	6		sem P	
DH-415	Ethics and law of AI	Rochel											2	2					4	100	sem A	sans retrait
CS-489	Experience design (pas donné en 24-25)	Huang						F					2		4				6		sem A	
CS-550	Formal verification	Kuncak	A		C	D							2	2	2				6		sem A	
CS-459	Foundations of probabilistic proofs	Chiesa				D					I		4	2					6		sem A	
CS-457	Geometric computing	Pauly						F					3		2				6		H	écrit
MATH-483	Gödel and recursivity ****	Duparc									I		2	2					5		H	écrit
CS-486	Interaction design	Pu						F				J				2	1	1	6		sem P	
MICRO-511	Image processing I	Unser/Van De Ville						F					3						3		H	écrit
MICRO-512	Image processing II	Unser/Van de Ville/Liebling/Sage						F								3			3		E	écrit
CS-487	Industrial automation	Tournier/Sommer														2		1	3	36	E	oral sans retrait
CS-430	Intelligent agents	Faltings										J	3	3					6		sem A	
CS-428	Interactive theorem proving	Barrière/Pit-Claudel			C						I					2	1	2	6		sem P	
CS-491	Introduction to IT consulting	Regev										J				6			6		E	oral
CS-431	Introduction to natural language processing	Chappelier/Rajman/Bosselut		B								J	2	2					6		H	écrit
CS-479	Learning in neural networks	Gerstner														2	1	1	6		E	oral
CS-526	Learning theory	Macris														2	2		6		E	écrit
CS-421	Machine learning for behavioral data ****	Käser														2		2	6		E	écrit
MGT-427	Management de projet et analyse du risque	Wieser											2		1				4		sem A	sans retrait
COM-516	Markov chains and algorithmic applications (pas donné en 24-25) ****	Lévêque/Macris		B							I		2	1	1				6		H	écrit
CS-552	Modern natural language processing	Bosselut		B								J				3	1	2	8		sem P	
EE-452	Networ machine learning	Frossard/Thanou														2		2	4		sem P	
COM-512	Networks out of control (pas donné en 24-25)	Thiran P./Grossglauser		B			E			H		J				2	1		6		E	écrit
MATH-489	Number theory II.c - Cryptography	Jetchev				D										2	2		5		E	écrit
CS-439	Optimization for machine learning	Flammarion														2	2	1	8		E	écrit
COM-507	Optional research project in Communication Systems	Divers enseignants											2						8		sem A ou P	
CS-522	Principles of computer systems	Argyraki/Candea	A		C	D			G				4						8		sem A	
CS-412	Software security	Payer			C	D			G							3	2	1	8		sem P	
MATH-486	Statistical mechanics and Gibbs measures (pas donné en 24-25)	Friedli														2	2		5		E	oral
PHYS-512	Statistical physics of computation	Erba											2	2					4		H	écrit
MATH-413	Statistics for Data Science	Chandaki/Limnios		B												4	2		8		E	écrit
COM-506	Student seminar: security protocols and applications	Vaudenay				D										2			3		sem P	
CS-448	Sublinear algorithms for big data analysis ****	Kapralov		B							I					2	1		6		sem P	
CS-460	Systems for data management and data science	Ailamaki/Kermarrec		B	C				G			J				2	2	2	8		E	écrit
CS-473	System programming for Systems-on-Chip	Kluter	A										2		2				6		sem A	
CS-510	Topics in software security	Payer				D							1	1					3		sem A	
CS-455	Topics in theoretical computer science (pas donné en 24-25) ****	Kapralov		B							I		3	1					6		sem A	
CS-444	Virtual reality	Boulic						F								2	1		6		sem P	
CS-503	Visual Intelligence : Machines and Minds	Zamir						F								2	2		6		sem P	

Remarques :

* Cf. l'art. 3 de l'Ordonnance sur le contrôle des études à l'EPFL

** sans retrait = pas de retrait possible après le délai d'inscription

**** cours biennaux donnés une année sur deux

2024-2025 **SYSTÈMES DE COMMUNICATION - Spécialisations**

Code	Matières	Enseignants sous réserve de modification	Crédits	Période des cours	
Spécialisation A "COMPUTER ENGINEERING"			**86**		
CS-420	Advanced compiler construction	Schinz	6		P
CS-470	Advanced computer architecture	Ienne	8		P
CS-471	Advanced multiprocessor architecture	Falsafi	8	A	
CS-477	Advanced operating systems	Kashyap	6	A	
EE-431 *	Advanced VLSI design	Burg/Levisse	4		P
CS-320 *	Computer language processing	Kuncak	6		P
CS-472	Design technologies for integrated systems	De Micheli	6	A	
ENG-466	Distributed intelligent systems	Martinoli	5	H	
CS-476	Embedded systems design	Kluter	6		P
CS-550	Formal verification	Kuncak	6	A	
EE-429 *	Fundamentals of VLSI Design	Burg	4	A	
EE-490(b) *	Lab in EDA based design	Koukab/Levisse	4	A	
EE-390(a) *	Lab on hardware-software digital systems codesign	Atienza/Peaon	3		P
CS-522	Principles of computer systems	Argyraki/Candea	8	A	
CS-473	System programming for Systems-on-chip	Kluter	6	A	
Spécialisation B "DATA ANALYTICS"			**134**		
CS-450	Algorithms II	Kapralov/Svensson	8	A	
COM-417	Advanced probability and applications	Shkel	8	A	
CS-401	Applied data analysis	Brbic	8	A	
CS-524	Computational complexity	Göös	6	A	
COM-480	Data visualization	Vuillon	6		P
CS-423	Distributed information systems	Aberer	6	A	
COM-404	Information theory and coding	Telatar	8	A	
COM-402	Information security and privacy	Payer	8	A	
COM-308 *	Internet analytics (pas donné en 24-25)	Grossglauser	6		P
CS-431	Introduction to natural language processing	Chappelier / Rajman/Bosselut	6	A	
COM-516	Markov chains and algorithmic applications (pas donné en 24-25)	Lévêque/Macris	6	A	
CS-552	Modern natural language processing	Bosselut	8		P
COM-512	Networks out of control (pas donné en 24-25)	Grossglauser/Thiran	6		P
CS-433	Machine learning	Jaggi/Flammarion	8	A	
COM-500	Statistical signal and data processing through applications	Ridolfi	8		P
MATH-413	Statistics for Data science	Chandaki/Limnios	8		P
CS-448	Sublinear algorithms for big data analysis	Kapralov	6		P
CS-460	Systems for data management and data science	Ailamaki/Kermarrec	8		P
CS-455	Topics in theoretical computer science (pas donné en 24-25)	Kapralov	6	A	
Spécialisation C "FOUNDATIONS OF SOFTWARE"			**84**		
CS-450	Algorithms II	Kapralov/Svensson	8	A	
CS-420	Advanced compiler construction	Schinz	6		P
CS-477	Advanced operating systems	Kashyap	6	A	
CS-524	Computational complexity	Göös	6	A	
CS-453	Concurrent computing	Guerraoui	6	A	
CS-451	Distributed algorithms	Guerraoui	8	A	
CS-550	Formal verification	Kuncak	6	A	
CS-452 *	Foundations of software	Bourgeat	8		P
CS-428	Interactive theorem proving	Barrière/Pit-Claudel	6		P
CS-522	Principles of computer systems	Argyraki/Candea	8	A	
CS-412	Software security	Payer	8		P
CS-460	Systems for data management and data science	Ailamaki/Kermarrec	8		P
Spécialisation D - CYBER SECURITY - SP			**115**		
CS-450	Algorithms II	Kapralov/Svensson	8	A	
CS-470	Advanced computer architecture	Ienne	8		P
COM-501	Advanced cryptography	Vaudenay	6		P
CS-477	Advanced operating systems	Kashyap	6	A	
CS-523	Advanced topics on privacy enhancing technologies	Troncoso	8		P
EE-431 *	Advanced VLSI design	Burg/Levisse	4		P
COM-401	Cryptography and security	Vaudenay	8	A	
CS-550	Formal verification	Kuncak	6	A	
CS-459	Foundations of probabilistic proofs	Chiesa	6	A	
EE-429 *	Fundamentals of VLSI Design	Burg	4	A	
COM-402	Information security and privacy	Payer	8	A	
COM-405	Mobile networks	Al Hassanieh	8	A	
MATH-489	Number theory II.c - Cryptography	Jetchev	5		P
CS-522	Principles of computer systems	Argyraki/Candea	8	A	
CS-412	Software security	Payer	8		P
COM-506	Student seminar : security protocols and applications	Vaudenay	3		P
COM-407	TCP/IP networking	Nikolopoulos	8	A	
CS-510	Topics in software security	Payer	3	A	

Légende :

* = cours hors plan d'études pour les étudiantes et étudiants ne faisant pas la spécialisation

2024-2025 **SYSTÈMES DE COMMUNICATION - Spécialisations**

Code	Matières		Enseignants sous réserve de modification	Crédits	Période des cours	
Spécialisation E. "NETWORKING AND MOBILITY"				**68**		
CS-450	Algorithms II		Kapralov/Svensson	8	A	
COM-401	Cryptography and security		Vaudenay	8	A	
CS-451	Distributed algorithms		Guerraoui	8	A	
CS-423	Distributed information systems		Aberer	6	A	
COM-402	Information security and privacy		Payer	8	A	
COM-405	Mobile networks		Al Hassanieh	8	A	
COM-430	Modern digital communication : A hands-on approach		Chiurtu	8	A	
COM-512	Networks out of control	(pas donné en 24-25)	Thiran P./Grossglauser	6		P
COM-407	TCP/IP networking		Nikolopoulos	8	A	
Spécialisation F. "SIGNALS, IMAGES, AND INTERFACES"				**125**		
CS-440	Advanced computer graphics	(pas donné en 24-25)	Jakob	6		P
EE-512	Applied biomedical signal processing		Lemay	4	A	
EE-554	Automatic speech processing		Magimai Doss	4	A	
BIO-410	Bioimage informatics		Sage/Seitz	4		P
CS-413	Computational photography		Süsstrunk	6		P
CS-341 *	Computer graphics		Pauly	6		P
CS-442	Computer vision		Fua	6		P
COM-418	Computers and music	(pas donné en 24-25)	Prandoni	6		P
COM-480	Data visualization		Vuillon	6		P
EE-559	Deep learning		Caballaro	4		P
CS-411	Digital education		Dillenbourg/Jermann	6		E
CS-489	Experience design	(pas donné en 24-25)	Huang	6	A	
CS-457	Geometric computing		Pauly	6	A	
EE-451 *	Image analysis and pattern recognition		Thiran J.-Ph.	4		P
MICRO-511	Image processing I		Unser/Van De Ville	3	A	
MICRO-512	Image processing II		Unser/Van de Ville/Liebling/Sage	3		P
CS-486	Interaction design		Pu	6		P
COM-430	Modern digital communication : A hands-on approach		Chiurtu	8	A	
CS-433	Machine learning		Jaggi/Flammarion	8	A	
EE-511 *	Sensors in medical instrumentation		A. Ionescu/Chételat	3		P
COM-500	Statistical signal and data processing through applications		Ridolfi	8		P
CS-444	Virtual reality		Boulic	6		P
CS-503	Visual intelligence : machines and minds		Zamir	6		P
Spécialisation G "SOFTWARE SYSTEMS"				**106**		
CS-420	Advanced compiler construction		Schinz	6		P
CS-470	Advanced computer architecture		Ienne	8		P
CS-471	Advanced multiprocessor architecture		Falsafi	8	A	
CS-477	Advanced operating systems		Kashyap	6	A	
CS-453	Concurrent computing		Guerraoui	6	A	
CS-438	Decentralized systems engineering		Borso/Ford	8	A	
CS-451	Distributed algorithms		Guerraoui	8	A	
CS-452 *	Foundations of software		Bourgeat	8		P
COM-402	Information security and privacy		Payer	8	A	
COM-405	Mobile networks		Al Hassanieh	8	A	
CS-522	Principles of computer systems		Argyraki/Candea	8	A	
CS-412	Software security		Payer	8		P
CS-460	Systems for data management and data science		Ailamaki/Kermarrec	8		P
COM-407	TCP/IP networking		Nikolopoulos	8	A	
Spécialisation H. "WIRELESS COMMUNICATIONS"				**54**		
COM-417	Advanced probability and applications		Shkel	8	A	
COM-404	Information theory and coding		Telatar	8	A	
COM-405	Mobile networks		Al Hassanieh	8	A	
COM-430	Modern digital communication : A hands-on approach		Chiurtu	8	A	
COM-512	Networks out of control	(pas donné en 24-25)	Thiran P./Grossglauser	6		P
COM-500	Statistical signal and data processing through applications		Ridolfi	8		P
COM-407	TCP/IP networking		Nikolopoulos	8	A	

Légende :

* = cours hors plan d'études pour les étudiantes et les étudiants ne faisant pas la spécialisation

2024-2025 **SYSTÈMES DE COMMUNICATION - Spécialisations**

Code	Matières		Enseignants sous réserve de modification	Crédits	Période des cours	
Spécialisation I. "COMPUTER SCIENCE THEORY"				**111**		
CS-450	Algorithms II		Kapralov/Svensson	8	A	
COM-417	Advanced probability and applications		Shkel	8	A	
CS-524	Computational complexity		Göös	6	A	
CS.453	Concurrent computing		Guerraoui	6	A	
CS-451	Distributed algorithms		Guerraoui	8	A	
CS-459	Foundations of probabilistic proofs		Chiesa	6	A	
CS-452	Foundations of software		Bourgeat	8		P
MATH-483	Gödel and recursivity		Duparc	5	A	
COM-404	Information theory and coding		Telatar	8	A	
CS-428	Interactive theorem proving		Barrière/Pit-Claudel	6		P
MATH-381 *	Logique mathématique		Duparc	5	A	
COM-300 *	Modèles stochastiques pour les communications		Thiran	6	A	
CS-433	Machine learning		Jaggi/Flammarion	8	A	
COM-516	Markov chains and algorithmic applications	(pas donné en 24-25)	Lévêque/Macris	6	A	
MATH-318 *	Set theory		Duparc	5		P
CS-448	Sublinear algorithms for big data analysis		Kapralov	6		P
CS-455	Topics in theoretical computer science	(pas donné en 24-25)	Kapralov	6	A	
Spécialisation J. "INTERNET INFORMATION SYSTEMS"				**76**		
COM-401	Cryptography and security		Vaudenay	8	A	
CS-451	Distributed algorithms		Guerraoui	8	A	
CS-423	Distributed information systems		Aberer	6	A	
CS-486	Interaction design		Pu	6		P
CS-430	Intelligent agents		Faltings	6	A	
CS-491	Introduction to IT consulting		Regev	6		P
CS-431	Introduction to natural language processing		Chappelier/Rajman/Bosselut	6	A	
CS-552	Modern natural language processing		Bosselut	8		P
COM-512	Networks out of control	(pas donné en 24-25)	Thiran P./Grossglauser	6		P
CS-433	Machine learning		Jaggi/Flammarion	8	A	
CS-460	Systems for data management and data science		Ailamaki/Kermarrec	8		P

Légende :

* = cours hors plan d'études pour les étudiantes et les étudiants ne faisant pas la spécialisation

RÈGLEMENT D'APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION DE SYSTÈMES DE COMMUNICATION

Année académique 2024-2025

du 20 juin 2024

La Vice-présidence académique,

vu l'ordonnance sur la formation menant au bachelor et au master de l'EPFL du 14 juin 2004,

vu l'ordonnance sur le contrôle des études menant au bachelor et au master à l'EPFL du 30 juin 2015,

vu le plan d'études de la section de Systèmes de communication,

arrête:

Article premier - Champ d'application

Le présent règlement fixe les règles d'application du contrôle des études de bachelor et de master en Systèmes de communication pour l'année académique 2024-2025.

Art. 2 – Étapes de formation

1 Le bachelor est composé de deux étapes successives de formation :

- le cycle propédeutique, d'une durée d'une année, dont la réussite se traduit par 60 crédits ECTS acquis en une fois, condition pour entrer au cycle bachelor. Le cycle propédeutique est commun avec celui de la section d'Informatique.

- le cycle bachelor, d'une durée de deux ans, dont la réussite implique l'acquisition de 120 crédits, condition pour entrer au master.

2 Le master effectué à l'EPFL est composé de deux étapes successives de formation :

- le cycle master, d'une durée de 3 semestres, dont la réussite implique l'acquisition de 90 crédits, condition pour effectuer le projet de master.

- le projet de master, d'une durée de 17 ou 25 semaines, dont la réussite se traduit par l'acquisition de 30 crédits.

Art. 3 – Sessions d'examens

1 Les branches de session sont examinées pendant les sessions d'hiver ou d'été (mention H ou E dans le plan d'études).

2 Les branches de semestre sont examinées pendant le semestre d'automne ou de printemps (mention sem A ou sem P).

3 Pour les branches de session, l'examen indiqué pour la session peut être complété par des contrôles de connaissances durant le semestre, selon les indications du personnel enseignant.

Chapitre 1 : Cycle propédeutique

Art. 4 - Examen propédeutique

1 L'examen propédeutique comprend deux blocs de branches :

- bloc 1 correspondant à 38 coefficients
- bloc 2 correspondant à 23 coefficients

2 L'examen propédeutique est réussi lorsque:

- à l'issue de la session d'hiver, une moyenne égale ou supérieure à 3.50 est atteinte dans le premier bloc, condition pour l'admission au semestre de printemps, et
- à l'issue de la session d'été, une moyenne égale ou supérieure à 4.00 est atteinte dans chacun des deux blocs, condition pour entrer au cycle bachelor.

3 En cas d’échec et répétition de l’examen propédeutique, les branches de semestre dont la note est égale ou supérieure à 4.00 ne peuvent pas être présentées une nouvelle fois.

Chapitre 2 : Cycle bachelor

Art. 5 - Régime transitoire

1 Les étudiantes et les étudiants qui ont débuté leur cycle bachelor avant le printemps 2023 restent soumis aux plan et règlement d’études initiaux et suivent les instructions de leur section concernant la répétition des branches échouées.

Art. 6 - Organisation

1 Les enseignements du bachelor sont répartis en quatre blocs et un groupe.

2 Le groupe “projets et options” est composé

- des cours de la liste du groupe “projets”
- des cours de la liste du groupe “options”
- des cours hors plan d’études suivant l’alinéa 3.

3 En 3e année, des cours pour un maximum de 12 crédits au total peuvent être choisis en dehors de la liste du plan d’études. Parmi ces 12 crédits, 6 crédits peuvent provenir d’un cours à options Master codifié CS ou COM. Tous les cours hors plan doivent obtenir l’approbation préalable de la direction de la section.

Art. 7 - Examens de 2e année

1 Le bloc A est réussi lorsque les **22 crédits** du plan d’études sont acquis.

2 Le bloc B est réussi lorsque les **24 crédits** du plan d’études sont acquis.

Art. 8 - Examens de 3e année

1 Le bloc C est réussi lorsque les **22 crédits** du plan d’études sont acquis.

2 Les **8 crédits** du groupe « projets » s’acquièrent de façon indépendante, par réussite individuelle.

Art. 9 - Examens sur les 2e et 3e années

1 Le bloc « SHS et MGT transversal » est réussi lorsque les **8 crédits** du plan d’études sont acquis.

2 Les **36 crédits** du groupe “options” s’acquièrent de façon indépendante, par réussite individuelle.

Chapitre 3 : Cycle master

Art. 10 - Organisation

1 Les étudiantes et les étudiants restent soumis aux plan et règlement d’études en vigueur lors de leur entrée au Master.

2 Les enseignements du cycle master sont répartis en un bloc et deux groupes. Ils peuvent donner lieu à l’acquisition d’une spécialisation ou d’un mineur.

3 Le bloc « Projets et SHS » est composé d’un projet de recherche et de l’enseignement SHS.

4 Le groupe 1 « Core courses » est composé des cours de la liste du plan d’études dans la rubrique « Master ».

5 Le groupe 2 « Options » est composé

- des cours de la liste du groupe 2 « options » du plan d'études dans la rubrique « Master » ;
- des crédits surnuméraires acquis dans le groupe 1 « Core courses » ;
- d'un projet de recherche optionnel de 8 crédits suivant l'alinéa 6 ;
- de cours hors plan d'études suivant l'alinéa 7;
- de cours liés à une spécialisation ou un mineur suivant l'art. 15.

6 Le projet de recherche du bloc « Projets et SHS » et le projet de recherche optionnel du groupe 2 ne peuvent pas être effectués durant le même semestre.

7 Des cours, comptant pour un maximum de 15 crédits au total, peuvent être choisis en dehors de la liste des cours sur le plan d'études dans la rubrique « Master ». Le choix de ces cours doit être approuvé préalablement par la direction de la section qui peut augmenter le maximum de 15 crédits si la demande est justifiée.

Art. 11 - Examens du cycle master

1 Le bloc « Projets et SHS » est réussi lorsque **18 crédits** sont acquis.

2 Le groupe « Core courses et Options », composé du groupe 1 « Core courses » et du groupe 2 « Options », est réussi lorsque **72 crédits** sont acquis.

3 Le groupe 1 « Core courses » est réussi lorsqu'**au moins 32 crédits** sont acquis.

Art. 12 - Enseignement SHS

1 L'enseignement SHS du semestre d'automne constitue l'introduction à la réalisation du projet SHS du semestre de printemps.

2 Pour autant qu'il considère que le cursus d'un cas individuel le justifie, le Collège des Humanités peut, d'entente avec l'équipe enseignante, déroger à cette organisation en autorisant que le projet soit réalisé au même semestre que le cours d'introduction ou soit réalisé à un semestre ultérieur.

Art. 13 – Mineurs et spécialisations

1 Afin d'approfondir un aspect particulier de sa formation ou de développer des interfaces avec d'autres sections, l'étudiante ou l'étudiant peut choisir la formation offerte dans le cadre d'un mineur figurant dans l'offre de l'EPFL ou d'une spécialisation de la section de Systèmes de communication.

2 Les mineurs « Data Science », « Informatique », « Cyber Security » et « Systèmes de Communication » ne peuvent pas être choisis.

3 L'étudiante ou l'étudiant annonce le choix d'un mineur à sa section au plus tard à la fin du premier semestre des études de master.

4 Le choix des cours qui composent un mineur se fait d'entente avec la section de Systèmes de communication et la personne responsable du mineur.

5 Le choix des cours qui composent une spécialisation est soumis à approbation de la section de Systèmes de communication.

6 L'étudiante ou l'étudiant qui choisit une spécialisation dans la liste figurant dans le plan d'études s'y inscrit au plus tard à la fin du premier semestre des études de master.

7 Un mineur ou une spécialisation est réussi quand 30 crédits au minimum sont acquis parmi les branches avalisées.

Chapitre 4 : Stage d'ingénierie

Art. 14 – Stage d'ingénierie

1 Un stage d'ingénierie doit être effectué dès la fin du 2e semestre et avant le projet de master. Sur demande, la section peut autoriser les titulaires d'un bachelor EPFL en Informatique ou Systèmes de communication, à réaliser le stage plus tôt.

2 Le stage prend l'une des formes suivantes :

- soit un stage d'été d'au minimum 8 semaines,

- soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre),

- soit un projet de master de 25 semaines en entreprise (art. 18).

3 Effectuer des cours/projet en parallèle au stage n'est pas autorisé.

4 La personne responsable des stages de la section évalue le stage, par l'appréciation « réussi » ou « échoué ». Sa réussite est une condition pour l'admission au projet de master. En cas d'échec, il peut être répété une fois, en règle générale dans une autre entreprise.

5 Il est validé avec les 30 crédits du projet de master.

6 Les modalités d'organisation et les critères de validation du stage font l'objet d'instructions internes à la section.

Chapitre 5 : Spécialisation en Informatique pour l'enseignement

Art. 15 – Procédure d'admission

1 Pour être admis à la spécialisation, la candidate ou le candidat doit être inscrit au Master en Systèmes de communication de l'EPFL et répondre aux conditions pour l'admission au Diplôme d'enseignement pour le degré secondaire II fixées par le Règlement d'application de la loi sur la HEP du 3 juin 2009 (RLHEP).

2 L'étudiante ou l'étudiant s'inscrit auprès de la HEP Vaud selon les conditions et délais de la candidature en ligne et transmet les pièces requises par le RLHEP, ainsi qu'une attestation d'immatriculation à l'EPFL.

Art. 16 – Organisation de la spécialisation

1 L'étudiante ou l'étudiant admis à cette spécialisation ne peut pas suivre d'autre spécialisation ou un mineur. Le plan d'études est modifié comme suit :

(i) Un nouveau groupe de 30 crédits de cours à la HEP Vaud est ajouté et le nombre de crédits du cycle master passe de 60 à 30 crédits ;
(ii) les cours SHS sont remplacés par un cours à la HEP Vaud ;
(iii) le projet de master peut s'étendre sur deux semestres et commencer après que le bloc «Projets et SHS » et le groupe « Core courses» aient été complétés;
(iv) la durée maximale des études complètes de master ne peut pas dépasser 8 semestres.

2 Au moins 50 crédits doivent avoir été acquis pour débuter la spécialisation.

Chapitre 6 : Projet de master

Art. 17 – Projet de master

1 Le projet de master s'étend sur une durée de 17 semaines s'il est effectué à l'EPFL ou de 25 semaines s'il est effectué hors EPFL.

2 Il est placé sous la responsabilité d'une professeure, d'un professeur ou MER affilié à la section d'Informatique ou de Systèmes de communication.

3 Sa réussite se traduit par l’acquisition de 30 crédits.

Chapitre 7 : Mobilité

Art. 18 – Périodes et conditions de mobilité

1 La section de Systèmes de communication offre la possibilité d’effectuer un séjour de mobilité en 3e année de bachelor et/ou dans le cadre du projet de master, aux conditions fixées ci-après.

2 Pour une mobilité en 3e année de bachelor,

- la moyenne atteinte à l’examen propédeutique doit s’élever au minimum à 4.50 (5.00 pour les échanges hors Europe);
- les 60 crédits de la 2e année doivent avoir été acquis.

En outre, des conditions spécifiques existant en fonction des destinations, l’accord de la personne déléguée à la mobilité est nécessaire.

3 L’admission conditionnelle au projet de master ne s’oppose en principe pas à une mobilité.

Au nom de l'EPFL

Le Vice-président académique, J. S. Hesthaven

Lausanne, le 20 juin 2024

CS-101

Advanced information, computation, communication I

Bourgeat Thomas, Käser Tanja

Cursus	Sem.	Type
Communication systems	BA1	Obl.
Computer science	BA1	Obl.

Language of teaching	English
Coefficient	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Remark

This course focuses on the foundational, discrete mathematics core of advanced computation.

Summary

Discrete mathematics is a discipline with applications to almost all areas of study. It provides a set of indispensable tools to computer science in particular. This course reviews (familiar) topics as diverse as mathematical reasoning, combinatorics, discrete structures & algorithmic thinking.

Content

I. Mathematical reasoning: propositional logic, propositional functions, quantifiers, rules of inference; this includes very basic logic circuits.
II. Sets and counting: cardinalities, inclusion/exclusion principle, sequences and summations.
III. Algorithms and complexity: basic algorithms, computational complexity, big-O notation and variants, countability.
IV. Number representations such as binary and hexadecimal and (postponed to 2nd semester) basic number theory: modular arithmetic, integer division, prime numbers, hash functions, pseudorandom number generation; applications.
V. Induction and recursion: mathematical induction, recursive definitions and algorithms.
VI. Basic combinatorial analysis: permutations, binomial theorem, counting using recursions.
VII. Basic probability: events, independence, random variables, Bayes' theorem.
VIII. Structure of sets: relations, equivalence relations, power set.

Keywords

Propositional logic, counting, complexity, big-O, number representations, sets, functions, relations, induction, basic probabilities, Bayes theorem, combinatorial analysis, recurrences, countability.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize if there is a mistake in a (simple) proof
- Apply general problem-solving techniques
- Recognize the mathematical structures present in applications
- Apply simple recursion and use it to design recursive algorithms
- Apply the tools studied in class to solve problems
- Demonstrate familiarity with mathematical reasoning
- Solve linear recurrences and use generating functions
- Argue about (un)countability
- Formulate complete, clear mathematical proofs

Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex cathedra lectures

Expected student activities

Studying the book, test your understanding by making the exercises, ask questions

Assessment methods

Continuous evaluations 10% and final exam 90%

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Additional Q&A sessions will take place on Tuesdays from 17:15-18:30 in INM 200 (starting in the second week of the semester)

Resources

Bibliography

"Discrete Mathematics and Its Applications", Kenneth H. Rosen, 8th ed, McGraw-Hill 2019. (You should be able to find the pdf on the web.)

Ressources en bibliothèque

- Discrete mathematics and its applications / Rosen

Websites

- http://will be provided later, if any

Moodle Link

- https://go.epfl.ch/CS-101

COM-102

Advanced information, computation, communication II

Gastpar Michael C.

Cursus	Sem.	Type
Communication systems	BA2	Obl.
Computer science	BA2	Obl.

Language of teaching	English
Coefficient	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

Text, sound, and images are examples of information sources stored in our computers and/or communicated over the Internet. How do we measure, compress, and protect the informatin they contain?

Content

I. How to measure information. Source modeling and probability. Entropy. Source coding and compression. Entropy to analyze algorithms.

II. Cryptography and information security. Modular arithmetic, modern algebra and number theory. The Chinese remainder theorem and RSA.

III. Protecting information. Channel modeling. A few finite fields. Vector spaces. Hamming distance. Linear codes. Reed-Solomon codes.

Keywords

Entropy
Data compression
Number theory
Cryptography
RSA cryptosystem
Linear codes
Reed-Solomon codes

Learning Outcomes

By the end of the course, the student must be able to:

- Understand Shannon's entropy
- Construct an optimal code
- Understand elementary number theory
- Know what an abelian group is
- Recognize a hidden isomorphism
- Know how RSA works
- Know a few linear codes on simple finite fields

Transversal skills

• Take feedback (critique) and respond in an appropriate manner.
• Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedrra with exercises

Expected student activities

Homework (written and grades) ever week.

Assessment methods

Continuous evaluations 10% and final exam 90%

Resources

Bibliography

"Sciences de l'information", J.-Y. Le Boudec, R. Urbanke et P. Thiran, online

Ressources en bibliothèque

- Introduction aux sciences de l'information : entropie, compression, chiffrement et correction d'erreurs / Le Boudec

Moodle Link

- https://go.epfl.ch/COM-102

MATH-111(a)

Algèbre linéaire

Basterrechea Sébastien

Cursus	Sem.	Type
Génie civil	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

L'objectif du cours est d'introduire les notions de base de l'algèbre linéaire et ses applications.

Contenu

1. Systèmes linéaires
2. Algèbre matricielle
3. Espaces vectoriels
4. Bases et dimension
5. Applications linéaires et matrices
6. Le déterminant d'une matrice
7. Valeurs propres, vecteurs propres, et diagonalisation
8. Produit scalaire
9. Matrices orthogonales et matrices symétriques.

Mots-clés

systèmes linéaires, espaces vectoriels, applications linéaires, matrices, déterminant, vecteurs propres, valeurs propres, orthogonalité, produit scalaire, décompositions matricielles, moindres carrés, chaînes de Markov

Compétences requises

Cours prérequis indicatifs

Cours de base.

Acquis de formation

- Effectuer des calculs standards en algèbre linéaire et en interpréter les résultats;
- Définir des concepts théoriques relevants de l'algèbre linéaire et en donner des exemples illustratifs;
- Identifier des exemples de concepts théoriques relevants de l'algèbre linéaire;
- Construire rigoureusement un raisonnement logique simple;
- Identifier quelques liens entre l'algèbre linéaire et d'autres branches des mathématiques.

Méthode d'enseignement

Cours en format de classe inversée, exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Bibliographie

Polycopié en ligne: botafogo.saitis.net/algebre-lineaire
Linear Algebra and its Applications, D.C. Lay (2nd or 3rd Addison-Wesley edition).
Linear Algebra and its Applications, D.C. Lay (3rd updated, 4th or new Pearson edition, 5th global Pearson edition).
Algèbre linéaire, théorie, exercices et applications, D.C. Lay (traduction 3e édition, de Boeck).
Algèbre linéaire et ses applications, D.C. lay (traduction 4e édition, Pearson).

Ressources en bibliothèque

- Linear Algebra and Its Applications (2nd and 3rd+updated) / Lay
- Linear Algebra and Its Applications (5th and 6th global edition) / Lay
- Algèbre linéaire : théorie, exercices & applications (trad. française, 2004)/ Lay
- Algèbre linéaire et applications (trad. française, 2012) / Lay

Polycopiés

En ligne.

Préparation pour

Analyse II, III et IV. Analyse numérique. Statistique.

MATH-111(en)

Linear algebra (english)

Iseli Annina

Cursus	Sem.	Type
Chemistry and chemical engineering	BA1	Obl.
Civil Engineering	BA1	Obl.
Communication systems	BA1	Obl.
Computer science	BA1	Obl.
Electrical and Electronical Engineering	BA1	Obl.
Environmental Sciences and Engineering	BA1	Obl.
Life Sciences Engineering	BA1	Obl.
Materials Science and Engineering	BA1	Obl.
Mechanical engineering	BA1	Obl.
Microtechnics	BA1	Obl.

Language of teaching	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	**228**

Summary

The purpose of the course is to introduce the basic notions of linear algebra and its applications.

Content

1. Linear systems;
2. Matrix algebra;
3. Vector spaces;
4. Bases and dimension;
5. Linear applications and matrices;
6. Determinant of a matrix;
7. Eigenvalues and eigenvectors;
8. Inner product, orthogonality, quadratic forms;
9. Orthogonal & Symmetric Matrices
10. Additional topics: 10.1 Singular value decomposition. 10.2. Systems of ODEs

Keywords

vector space, linearity, matrix, determinant, orthogonality, inner product

Learning Outcomes

By the end of the course, the student must be able to:

- Accurately make standard computations relevant to linear algebra and interpret the results;
- Define and provide illustrative examples of relevant theoretical notions;
- Identify examples of relevant theoretical notions;
- Construct a simple logical argument rigorously;
- Identify some connections between linear algebra and other branches of mathematics.

Teaching methods

Lectures and exercises in the classroom

Assessment methods

Written exam

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

Linear Algebra and its Applications / D.C. Lay etal, preferably 5th edition

Ressources en bibliothèque

- Linear Algebra and its Applications / Lay

Moodle Link

- https://go.epfl.ch/MATH-111_en

Videos

- http://No videos. Lectures only take place in person in the class room.

Prerequisite for

Analysis II, III and IV, Numerical Analysis Statistics

MATH-111(pi)

Algèbre linéaire (classe inversée)

Deparis Simone

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Ingénierie des sciences du vivant	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	**194**

Remarque

Cours en classe inversée, merci de consulter https://go.epfl.ch/classes-inversees avant de vous inscrire

Résumé

L'objectif du cours est d'introduire les notions de base de l'algèbre linéaire et ses applications. Cette classe pilote est donné sous forme inversée.

Contenu

1. Systèmes linéaires
2. Algèbre matricielle
3. Espaces vectoriels
4. Bases et dimension
5. Applications linéaires et matrices
6. Le déterminant d'une matrice
7. Valeurs propres, vecteurs propres, et diagonalisation
8. Produits scalaires et espaces euclidiens
9. Matrices orthogonales et matrices symétriques

Mots-clés

espace vectoriel, linéarite, matrice, déterminant, orthogonalité, produit scalaire

Compétences requises

Cours prérequis indicatifs

cours de base

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Effectuer des calculs standards en algèbre linéaire et en interpréter les résultats;
- Définir des concepts théoriques relevants de l'algèbre linéaire et en donner des exemples illustratifs;
- Identifier des exemples de concepts théoriques relevants de l'algèbre linéaire;
- Construire rigoureusement un raisonnement logique simple;
- Identifier quelques liens entre l'algèbre linéaire et d'autres branches des mathématiques.

Méthode d'enseignement

Cours ex cathedra, exercices en salle.
Le cours est sous forme classe inversée. L'étudiant devra se préparer aux séances en classe en suivant le cours sur le MOOC.

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non
Autres	RAQ supplémentaires avec l'enseignant

Ressources

Bibliographie

Algèbre linéaire et applications, David C. Lay, 5e edition, editeur: Pearson, ISBN 978-2-7613-9109-2 (pas besoin de MonLab)
Algèbre linéaire et applications, David C. Lay, 4e edition, editeur: Pearson, ISBN: 978-2-7440-7583-4

Ressources en bibliothèque

- Algèbre linéaire et applications / Lay

Sites web

- https://courseware.epfl.ch/courses/course-v1:EPFL+AlgebreLineaire+2018

Liens Moodle

- https://go.epfl.ch/MATH-111_pi

Préparation pour

Suite des études en ingénierie et sciences.

MATH-101(a) **Analyse I**

Mila Olivier

Cursus	Sem.	Type
Génie civil	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Étudier les concepts fondamentaux d'analyse et le calcul différentiel et intégral des fonctions réelles d'une variable.

Contenu

- Raisonner, démontrer et argumenter en mathématiques
- Nombres, structures et fonctions
- Suites, limites et continuité
- Séries numériques
- Fonctions réelles et processus de limite
- Calcul différentiel et intégral

Mots-clés

nombres réels, fonction, suite numérique, suite convergente/divergente, limite d'une suite, sous-suite, fonction, limite d'une fonction, fonction continue, série numérique, série convergente/divergente, convergence absolue, dérivée, classe C^k, théorème(s) des accroissements finis, développement limité, série entière, intégrale de Riemann, primitive, théorème de la valeur moyenne

Acquis de formation

- Le but fondamental de ce cours est d'acquérir les compétences suivantes :
- Raisonner rigoureusement pour analyser des problèmes
- Choisir ou sélectionner les outils d'analyse pertinents pour résoudre des problèmes
- Identifier les concepts inhérents à chaque problème
- Appliquer efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Résoudre les problèmes de convergence, de suites et de séries
- Maîtriser les techniques du calcul différentiel et intégral
- Parmi les outils de base, on trouve les notions de convergence, de suites et de séries. Les fonctions d'une variable seront étudiées rigoureusement, avec pour but une compréhension approfondie des techniques du calcul différentiel et intégral.

Méthode d'enseignement

Cours ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui
Autres	Tutorat des exercices

Ressources

Service de cours virtuels (VDI)

Non

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. PPUR.

Ressources en bibliothèque

- Calcul différentiel et intégral (en ligne)
- Calcul différentiel et intégral (livres) / Douchet & Zwahlen

Sites web

- http://moodle.epfl.ch/

Liens Moodle

- https://go.epfl.ch/MATH-101_a

MATH-101(de) Analyse I (allemand)

Krieger Joachim

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Ingénierie des sciences du vivant	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	allemand
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	**77**

Résumé

Es werden die Grundlagen der Analysis sowie der Differential- und Integralrechnung von Funktionen einer reellen Veränderlichen erarbeitet.

Contenu

- Grundlagen für Mathematisches Begründen, Beweisen, und Argumentieren

- Körper-/Anordnungsaxiome, reelle Zahlen

- Funktionen

- Folgen, Grenzwerte und Stetigkeit

- Reihen

- Reelle Funktionen und Grenzwerte

- Differential- und Integralrechnung

Mots-clés

Funktionen, Folge, konvergente/divergente Folge, Grenzwert einer Folge, Teilfolge, Grenzwert einer Funktion, stetige Funktion, Reihe, konvergente/divergente Reihe, absolute Konvergenz, Ableitung, Funktionsklasse C^k, Mittelwertsatz der Differentialrechnung, Taylor-Entwicklung, Potenzreihe, Riemann-Integral, Stammfunktion, Mittelwertsatz der Integralrechnung

Acquis de formation

- Raisonner rigoureusement pour analyser des problèmes
- Choisir ou sélectionner les outils d'analyse pertinents pour résoudre des problèmes
- Identifier les concepts inhérents à chaque problème
- Appliquer efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Résoudre les problèmes de convergence, de suites et de séries
- Analyser des problèmes nouveaux

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.
- Gérer ses priorités.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.

Méthode d'enseignement

Vorlesungen und Tutorien

Méthode d'évaluation

Schriftliche Klausur

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Non

Polycopiés

Ein ergänzendes deutschsprachiges Vorlesungsskript wird zur Verfügung gestellt.

Liens Moodle

- https://go.epfl.ch/MATH-101_de

MATH-101(en)

Analysis I (English)

Mountford Thomas

Cursus	Sem.	Type
Chemistry and chemical engineering	BA1	Obl.
Civil Engineering	BA1	Obl.
Communication systems	BA1	Obl.
Computer science	BA1	Obl.
Electrical and Electronical Engineering	BA1	Obl.
Environmental Sciences and Engineering	BA1	Obl.
Life Sciences Engineering	BA1	Obl.
Materials Science and Engineering	BA1	Obl.
Mechanical engineering	BA1	Obl.
Microtechnics	BA1	Obl.

Language of teaching	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	**216**

Summary

We study the fundamental concepts of analysis, calculus and the integral of real-valued functions of a real variable.

Content

- Reasoning, proving and arguing in mathematics
- Numbers, structures and functions
- Sequences, limit and continuity
- Series of reals
- Real-valued functions of a real variable and convergence
- Differential Calculus and the Integral

Keywords

Real numbers, function, sequence,convergent/divergent sequence, limit, subsequence, limit of a function, continuous function, series of real numbers, convergent/divergent series, absolute convergence, derivative, class C^k, mean value theorem, Taylor's theorem, Taylor series, Riemann integral, indefinite integral, intermediate value theorem.

Learning Outcomes

- The intended learning outcomes of this course are that students acquire the following capacities:
- Reason rigorously to analyse problems
- Choose appropriate analytical tools for problem solving.
- Be able to conceptualise in view of the applications of analysis.
- Apply efficiently mathematical concepts for problem solving by means of examples and exercises
- Analyze and to solve new problems.
- Master the basic tools of analysis as, for example, notions of convergence, sequences and series.
- Studying rigorously real functions we intend that students will demonstrate a deep understanding of calculus

Teaching methods

Ex cathedra/online lectures and exercise sessions with tutors and student assistants.

Assessment methods

Written exam

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Tutoring of exercises other measures to be defined

Resources

Moodle Link

- https://go.epfl.ch/MATH-101_en

EPFL

MATH-101(pi) Analyse I (classe inversée)

Friedli Sacha

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Ingénierie des sciences du vivant	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	**194**

Remarque

Cours en classe inversée, merci de consulter https://go.epfl.ch/classes-inversees avant de vous inscrire

Résumé

Étudier les concepts fondamentaux d'analyse et le calcul différentiel et intégral des fonctions réelles d'une variable. Cette classe est donnée sous forme inversée.

Contenu

- Raisonner, démontrer et argumenter en mathématiques
- Nombres, structures et fonctions
- Suites, limites et continuité
- Séries numériques
- Fonctions réelles et processus de limite
- Calcul différentiel et intégral

Mots-clés

nombres réels, suites numériques, suites convergentes/divergentes, limite d'une suite, sous-suites, fonctions, limite d'une fonction, fonctions continues, séries numériques, séries convergentes/divergentes, convergence absolue, dérivée, classe C^k, théorème(s) des accroissements finis, développement limité, séries entières, intégrale de Riemann, primitives, théorème de la valeur moyenne.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Le but fondamental de ce cours est d'acquérir les compétences suivantes :
- Raisonner rigoureusement pour analyser des problèmes
- Choisir ou sélectionner les outils d'analyse pertinents pour résoudre des problèmes
- Identifier les concepts inhérents à chaque problème
- Appliquer efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Résoudre les problèmes de convergence, de suites et de séries
- Maîtriser les techniques du calcul différentiel et intégral
- Parmi les outils de base, on trouve les notions de convergence, de suites et de séries. Les fonctions d'une variable

seront étudiées rigoureusement, avec pour but une compréhension approfondie des techniques du calcul différentiel et intégral.

Méthode d'enseignement

Cours ex cathedra, exercices en salle.
Le cours est sous forme classe inversée. L'étudiant·e devra se préparer aux séances de contact.
Plus d'informations sur botafogo.saitis.net/analyse-1-flipped

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui
Autres	Tutorat des exercices. Portail Moodle.

Ressources

Polycopiés

botafogo.saitis.net/analyse-1

Sites web

- http://botafogo.saitis.net/analyse-1-flipped

Préparation pour

Analyse II

EPFL

MATH-106(e) **Analyse II**

Lachowska Anna

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Étudier les concepts fondamentaux d'analyse et le calcul différentiel et intégral des fonctions réelles de plusieurs variables.

Contenu

- L'espace R^n
- Calcul différentiel des fonctions à plusieurs variables
- Intégrales multiples
- Équations différentielles ordinaires
- Méthodes de démonstration et arguments mathématiques

Mots-clés

Espace vectoriel euclidien, , dérivée partielle, différentielle, matrice jacobienne, extremum local d'une fonction de plusieurs variables, matrice hessienne, développement limité, gradient, divergence, rotationnel, règle de composition, théorème des fonctions implicites, multiplicateurs de Lagrange, intégrale multiple, équation différentielle ordinaire

Compétences requises

Cours prérequis obligatoires

Analyse I, Algèbre linéaire I

Cours prérequis indicatifs

Analyse I, Algèbre linéaire I

Concepts importants à maîtriser

- calcul différentiel et intégral des fonctions réelles d'une variable
- les notions de convergence
- espace vectoriel, matrices, valeurs propres

Acquis de formation

- Le but fondamental de ce cours reste, comme pour la partie I, d'acquérir les capacités suivantes :
- Appliquer
- avec aisance et approfondir les compétences et connaissances acquises en Analyse I :
- Raisonner

- rigoureusement pour analyser les problèmes
- Choisir ou sélectionner
- les outils d'analyse pertinents pour résoudre des problèmes
- Identifier
- les concepts inhérents à chaque problème
- Appliquer
- efficacement les concepts pour résoudre les exercices similaires aux exemples et exercices traités au cours
- Se montrer capable d'analyser et de résoudre des problèmes nouveaux
- Maîtriser les techniques du calcul différentiel et intégral.
- Maîtriser les équations différentielles élémentaires, l'espace R^n, les fonctions de plusieurs variables, les dérivées partielles et les intégrales multiples.

Méthode d'enseignement

Cours ex cathedra et exercices en salle

Méthode d'évaluation

Examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui
Autres	Tutorat des exercices autres mesures à définir

Ressources

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. PPUR, 2011.
L'enseignant précisera les manuels recommandés dans son cours.

Ressources en bibliothèque

- Calcul différentiel et intégral / Douchet et Zwahlen

Liens Moodle

- https://go.epfl.ch/MATH-106_e

MATH-106(en)

Analysis II (English)

Richter Florian Karl

Cursus	Sem.	Type
Chemistry and chemical engineering	BA2	Obl.
Civil Engineering	BA2	Obl.
Communication systems	BA2	Obl.
Computer science	BA2	Obl.
Electrical and Electronical Engineering	BA2	Obl.
Environmental Sciences and Engineering	BA2	Obl.
Life Sciences Engineering	BA2	Obl.
Materials Science and Engineering	BA2	Obl.
Mechanical engineering	BA2	Obl.
Microtechnics	BA2	Obl.

Language of teaching	English
Coefficient	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	**230**

Summary

The course studies fundamental concepts of analysis and the calculus of functions of several variables.

Content

-The Euclidean space R^n.
-Vector functions and curves
-Differentiation of functions of several variables.
-Multiple integrals
-Ordinary differential equations.

Keywords

Euclidean vector space, partial derivative,differential, Jacobian, Hessian, Taylor expansion, gradient, chain rule, implicit function theorem, Lagrange multipliers, multiple integrals, ordinary differential equation

Learning Prerequisites

Required courses

Analysis I, Linear Algebra I

Important concepts to start the course

-

Learning Outcomes

- The goal of this course consists as for Analysis 1 is that students acquire the following capacities:
- Consolidate the skills and knowledge they acquired in Analysis 1.
- Reason
- rigorously and to analyse problems
- Choose
- appropriate analytical tools for problem solving.
- Conceptualize problems
- Apply

- efficiently mathematical concepts for problem solving by means of examples and exercises
- Analyze
- and to solve new problems.
- Master the basic tools of analysis
- Master the basic tools of elementary ordinary differential equations, the Euclidean space R^n and functions of several variables
- Apply the skills and knowledge they acquired in Analysis 1.
- Reason rigorously and analyse problems
- Develop appropriate analytical tools for problem solving.
- Analyze efficiently mathematical concepts for problem solving by means of examples and exercises

Teaching methods

lectures, exercises sessions in the classroom.

Assessment methods

Written exam

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. PPUR, 2011.

Ressources en bibliothèque

- Calcul différentiel et intégral / Douchet et Zwahlen

Moodle Link

- https://go.epfl.ch/MATH-106_en

CS-173

Fundamentals of digital systems

Stojilovic Mirjana

Cursus	Sem.	Type
Communication systems	BA2	Obl.
Computer science minor	E	Opt.
Computer science	BA2	Obl.

Language of teaching	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	4 weekly
Number of positions	

Summary

Welcome to the introductory course in digital design and computer architecture. In this course, we will embark on a journey into the world of digital systems, exploring the fundamental principles and concepts that underpin modern computing technology.

Content

Part I: Number Systems
Digital systems are built from circuits that process binary digits (zeros and ones). This part of the course aims to show how familiar numeric quantities can be represented and manipulated in a digital system. We will describe binary number systems and show how addition and subtraction are performed.

Part II: Digital Logic and Design with Verilog
Binary logic circuits have the dominant role in digital technology. We will discover logic gates and expressions, combinational and sequential logic elements, and build state machines. We will design and simulate logic circuits in Logisim-Evolution and Verilog hardware description language.

Part III: Intro to Computer Architecture
We will discover the fundamental building blocks of computers and how they are organized. On one side, we will study a basic processor architecture and the RISC-V open-standard ISA. On the other, we will discuss the future of computing and the role of domain-specific architectures. The basics of FPGA architecture and design flow will also be covered.

Keywords

- number systems, logic gates, digital logic, delay, frequency, power, arithmetic circuits, combinational circuits, sequential circuits, finite state machines
- memory, computer organization, CPU, FPGA, GPU, TPU
- RISC-V ISA
- Logisim-Evolution, Verilog, Icarus Verilog, GTKWave, Visual Studio Code

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze the behavior of a digital circuit
- Describe digital systems in Verilog
- Design and test digital systems

- Explain basic components of a computer
- Create fully functional digital systems, including a simple CPU
- Realize a fully functional piece of RISC-V assembly code
- Describe the functionality of a piece of RISC-V assembly code
- Characterize various number systems
- Reason about the factors affecting the performance of digital systems
- Describe the importance of reduced-precision computing
- Design common arithmetic circuits

Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Communicate effectively, being understood, including across different languages and cultures.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools

Teaching methods

- Ex cathedra
- Flipped classroom
- Exercices

Expected student activities

- Attending the course and exercise sessions
- Completing the assignments
- Participating in the discussions on the forum

Assessment methods

- Midterm exam (written)
- Final exam in the exam session (written)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

- Stephen Brown and Zvonko Vranesic (2014), *Fundamentals of Digital Logic with Verilog Design* (3rd ed.), McGraw-Hill

- John F. Wakerly (2018), *Digital Design Principles and Practices with Verilog* (5th ed.), Pearson
- Milos D. Ercegovac and Tomas Lang (2004), *Digital Arithmetic*, Morgan Kauffmann Publishers
- David Patterson and John Hennessy (2020), *Computer Organization and Design RISC-V Edition: The Hardware Software Interface* (2nd ed.), Morgan Kauffmann Publishers
- David Patterson and Andrew Waterman (2017), *The RISC-V Reader: An Open Architecture Atlas* (1st ed.), Strawberry Canyon LLC

Ressources en bibliothèque

- Fundamentals of Digital Logic with Verilog Design / Brown
- Digital Arithmetic / Ercegovac
- Digital Design Principles and Practices with Verilog / Wakerly
- Computer Organization and Design RISC-V Edition / Patterson
- The RISC-V Reader / Patterson

Websites

- http://mirjanastojilovic.github.io/cs173

Moodle Link

- https://go.epfl.ch/CS-173

Videos

- https://mediaspace.epfl.ch/channel/CS-173%2BFundamentals%2Bof%2BDigital%2BSystems/66670

Prerequisite for

CS-200 Computer Architecture

EPFL

CS-107 Introduction à la programmation

Pereira Pires Rafael, Sam Jamila

Cursus	Sem.	Type
Auditeurs en ligne	H	Opt.
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	5
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Labo	1 hebdo
Nombre de places	

Résumé

Ce cours aborde les concepts fondamentaux de la programmation et de la programmation orientée objet (langage JAVA). Il permet également de se familiarier avec un environnement de développement informatique (par défaut sous Unix).

Contenu

- Prise en main d'un environnement de programmation (éditeur, compilateur, ...).

- Initiation à la programmation (langage JAVA) : variables, expressions, structures de contrôle, modularisation, entrées-sorties

- Introduction à la programmation objet (langage JAVA) : objets, classes, méthodes, abstraction, encapsulation, héritage, polymorphisme

- Pratique de concepts algorithmques fondamentaux (récursion, recherche, tri etc.).

Mots-clés

Java, programmation orientée-objet, Unix.

Compétences requises

Cours prérequis obligatoires

Aucun

Cours prérequis indicatifs

Aucun

Concepts importants à maîtriser

Aucun

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Concevoir des algorithmes résolvant des tâches simples
- Transcrire un algorithme en son programme équivalent en Java
- Modéliser en langage Java une situation simple du monde réelle
- Structurer un problème complexe en sous-problèmes
- Analyser un code pour en décrire le résultat ou le corriger
- Argumenter la validité de décision de conception de base dans un programme orienté-objet
- Tester l'adéquation du résultat d'un programme par rapport à la tâche visée
- Réaliser de façon autonome une application de petite taille au moyen du langage Java et en utilisant les concepts fondamentaux de la programmation orientée objet
- Concevoir des algorithmes résolvant des tâches simples
- Transcrire un algorithme en son programme équivalent en Java
- Modéliser en langage Java une situation simple du monde réel
- Structurer un problème complexe en sous-problèmes
- Analyser un code pour en décrire le résultat ou le corriger
- Argumenter la validité de décision de conception de base dans un programme orienté-objet
- Tester l'adéquation du résultat d'un programme par rapport à la tâche visée
- Réaliser de façon autonome une application de petite taille au moyen du lanage Java et en utilisant les concepts fondamentaux de la programmation orientée objet

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Accéder aux sources d'informations appropriées et les évaluer.

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur et support en ligne MOOC

Travail attendu

participation au cours, résolutions d'exercices.

Méthode d'évaluation

1. Examen écrit individuel (40%)
2. Mini-projet 1 auto-évalué (non noté)
3. Mini-projet 2 (60%)

Les mini-projets se font à deux.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Oui

Bibliographie

Notes de cours disponibles en ligne. Livre(s) de référence indiqué(s) en début de semestre

Liens Moodle

- https://go.epfl.ch/CS-107

Vidéos

- https://www.coursera.org/learn/initiation-programmation-java/
- https://www.coursera.org/learn/programmation-orientee-objet-java

Préparation pour

Pratique de la programmation orientée-objet (CS-108)

PHYS-101(a)

Physique générale : mécanique

Hogge Jean-Philippe

Cursus	Sem.	Type
Génie civil	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Le but du cours de physique générale est de donner à l'étudiant les notions de base nécessaires à la compréhension des phénomènes physiques. L'objectif est atteint lorsque l'étudiant est capable de prévoir quantitativement les conséquences de ces phénomènes avec des outils théoriques appropriés.

Contenu

Introduction et Cinématique : référentiels, trajectoire, vitesse, accélération, coordonnées cartésiennes et cylindriques.
Dynamique du point matériel : quantité de mouvement, lois de Newton, forces fondamentales, empiriques et de liaison, mouvement oscillatoire, moment cinétique.
Travail, puissance, énergie : énergies cinétique, potentielle, mécanique, lois de conservation, mouvements gravitationnels, collisions.
Changement de référentiels : dynamique dans les référentiels non inertiels
Dynamique des systèmes : centre de masse, moment cinétique, énergie
Solide indéformable : moment cinétique, moment d'inertie, effets gyroscopiques

Compléments
L'enseignement peut contenir, mais pas exclusivement, les éléments suivants: mécanique analytique, coordonnées sphériques, relativité restreinte

Mots-clés

Physique générale, mécanique du point matériel, mécanique du solide, coordonnées, cinématique, relativité, énergie, travail

Compétences requises

Cours prérequis indicatifs

- Niveau mathématique de la maturité fédérale, voir par exemple "www.vsmp.ch/crm/cat.htm"
- "Savoir-Faire en Maths - bien commencer ses études scientifiques", Y. Biollay, A. Chaabouni, J. Stubbe, PPUR, 2010

Concepts importants à maîtriser

Espace vectoriel, produit scalaire et produit vectoriel, dérivation et intégration d’une fonction réelle, équations différentielles ordinaires

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Elaborer un modèle physique d'un système mécanique
- Démontrer un savoir-faire dans la résolution de problèmes
- Structurer les modèles en termes d'équations différentielles
- Formuler et utiliser des hypothèses simplificatrices pour décrire une expérience
- Utiliser les modèles théoriques qui décrivent la Nature
- Estimer les ordres de grandeur
- Relier les notions de cours et les observations du monde quotidien

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Cours, exercices en salle et travail personnel

Méthode d'évaluation

Examen écrit à la session d'hiver

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Service de cours virtuels (VDI)

Non

Bibliographie

Livre de cours:
- "Mécanique", J.-Ph. Ansermet, Presses polytechniques universitaires romandes, 2009, ISBN 978-2-88074-829-6

MOOC:
- "Mécanique", J.-Ph. Ansermet, www.coursera.org/course/mecanique, 2013

Autres références:
- "Mécanique", Berkeley, cours de physique Vol. 1

Ressources en bibliothèque

- La Mécanique / Ansermet

Polycopiés

Copie des transparents et autres ressources disponibles sur le site web du cours dans moodle.

Sites web

- http://moodle.epfl.ch/course/view.php?id=14244

Liens Moodle

- https://go.epfl.ch/PHYS-101_a

Préparation pour

Physique générale II

PHYS-101(de)

Physique générale : mécanique (allemand)

Gruetter Rolf

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Ingénierie des sciences du vivant	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	allemand
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	**257**

Résumé

Die Studenten erwerben die Grundkenntnisse zum Verständnis von physikalischen Phänomenen der Mechanik. Sie entwickeln die Fähigkeit, die Auswirkungen dieser Phänomene mit den entsprechenden theoretischen Werkzeugen quantitativ zu bestimmen.

Contenu

Enleitung und Kinematik: Abschätzen von Grössenordnungen, physikalische Denkweise, Bezugssysteme, Bahnkurve, Geschwindigkeit, Beschleunigung, kartesische und zylindrische Koordinatensysteme
Dynamik des Massenpunktes: Impuls, Newton's Axiome, grundlegende und empirische Kräfte(verhältnisse), Mechanische Schwingungen, Drehimpuls.
Arbeit, Leistung, Energie: Kinetische, potentielle und mechanische Energie, Erhaltungssätze, Gravitation. Kollisionen.
Aenderung der Bezugssysteme: beschleunigte Bezugssysteme
Mechanik von Systemen: Massenzentrum, (Dreh)impuls, Energie
Mechanik starrer Körper: Drehimpuls, Trägheitsimpuls, Hebelgesetz, gyroskopische Effekte
Ergänzungen: Der Stoff kann folgende nicht-inklusive Elemente beinhalten: sphärische Koordinatensystem, Einführung in die spezielle Relativitätstheorie

Mots-clés

Allgemeine Physik, Koordinaten, Kinematik, Energie, Arbeit, Mechanik des starren Körpers, Koordinaten, Relativität.

Compétences requises

Cours prérequis indicatifs

Ausgezeichnete Grundkenntnisse der Mathematik Niveau Schweizerische Maturitätsprüfung (zB. http://www.math.ch/kanon/)

Concepts importants à maîtriser

Vektoralgebra: Skalar- und Vektorprodukt, Zerlegen von Vektoren. Beziehungen des rechtwinkligen Dreiecks.
Lösung von linearen Gleichungsystemen mit 2 oder 3 Unbekannten.
Integration/Differentiation von Funktionen und Vektoren.
Umwandlung physikalischer Einheiten

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Etablir eines physikalischen Modells eines mechanischen Systems.
- Reconnaitre eines physikalischen Modells eines mechanischen
- Développer der korrekten Einheiten
- Juger benutzter vereinfachender Annahmen
- Identifier des vorhergesehenen qualtitativen Verhaltens
- Estimer von Größenordnungen
- Reconnaitre Zusammenhänge zwischen Vorlesung und Alltag
- Mettre en ordre der signifikanten Stellen
- Etablir Herleiten der Bewegungsgleichungen

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Kurs, Ubungen im Saal und persönliche Arbeit.

Travail attendu

Neben der Vorlesung und Uebungs-sessionen (Diese Zeitangaben können von der geleisteten Vorbereitung aufs Studium und der Qualität der Arbeitsorganisation abhangen):
Vor der Vorlesung, ca. 2 bis 3 Seiten im Vorlesungswerk lesen (ca. 15 min)
Nach der Vorlesung, eine Zusammenfassung/Formelsammlung erstellen (ca. 30 min.), gefolt von mehreren Vorbereitungsübungen (ca. 60 min)

Méthode d'évaluation

schriftliche Prüfung

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Ressources

Service de cours virtuels (VDI)

Non

Bibliographie

- Physics for scientists and engineers, 5ème édition, Giancoli. International Edition, Pearson/Prentice Hall
- Mathematics for physics with calculus, Das, Pearson/Prentice Hall

- W. Demtröder, Experimentalphysik 1, Springer Verlag

Ressources en bibliothèque

- Physics for scientists and engineers / Giancoli
- Experimentalphysik 1 / Demtröder
- Mathematics for physics with calculus / Das

Sites web

- http://lifmet.epfl.ch

Liens Moodle

- https://go.epfl.ch/PHYS-101_de

Préparation pour

Physique générale - thermodynamique

EPFL

PHYS-101(en) **General physics : mechanics (English)**

Baquero Ruiz Marcelo

Cursus	Sem.	Type
Chemistry and chemical engineering	BA1	Obl.
Civil Engineering	BA1	Obl.
Communication systems	BA1	Obl.
Computer science	BA1	Obl.
Electrical and Electronical Engineering	BA1	Obl.
Environmental Sciences and Engineering	BA1	Obl.
Life Sciences Engineering	BA1	Obl.
Materials Science and Engineering	BA1	Obl.
Mechanical engineering	BA1	Obl.
Microtechnics	BA1	Obl.

Language of teaching	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	**370**

Summary

Students will learn the principles of mechanics to enable a better understanding of physical phenomena, such as the kinematics and dyamics of point masses and solid bodies. Students will acquire the capacity to quantitatively analyze these effects with the appropriate theoretical tools.

Content

The course contains the following elements (not exclusively):

Mechanics

Introduction and kinematics
Reference frames, trajectories, velocity, acceleration, and Cartesian, spherical, and cylindrical coordinates.

Dynamics of the point mass and solid body
Momentum, Newton's laws, fundamental forces, empirical forces, constraints, oscillatory motion, and angular momentum.

Work, power, energy
Kinetic energy, potential energy, conservation laws, gravitational motion, collisions.

Keywords

General physics, point masses, coordinates, kinematics, energy, work

Learning Prerequisites

Recommended courses

Math level required for "maturité fédérale", which indicates the level of math appropriate for a good start at EPFL.

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate models of basic physical systems
- Structure models in terms of differential equations
- Apply simplifying assumptions to describe a phenomenon

- Estimate orders of magnitude
- Distinguish the theoretical models describing nature
- Contextualise theoretical models in every day life
- Develop the know-how to solve a problem

Transversal skills

- Use a work methodology appropriate to the task.

Teaching methods

Lectures and exercises

Assessment methods

The course concludes with a written exam

Resources

Bibliography

- Serway, Physics for Scientists and Engineers.
- Douglas Giancoli. Physics for Scientists and Engineers. 4th Edition.
- D. Halliday, R. Resnick, K. S. Krane. Physics, Volume 1.

Ressources en bibliothèque

- Serway, Physics for Scientists and Engineers.
- D. Halliday, R. Resnick, K. S. Krane. Physics, Volume 1
- Douglas Giancoli. Physics for Scientists and Engineers. 4th Edition

Moodle Link

- https://go.epfl.ch/PHYS-101_en

Prerequisite for

General physics II

PHYS-101(l)

Physique générale : mécanique (classe inversée)

Hébert Cécile

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Génie civil	BA1	Obl.
Génie mécanique	BA1	Obl.
Génie électrique et électronique	BA1	Obl.
Informatique	BA1	Obl.
Ingénierie des sciences du vivant	BA1	Obl.
Microtechnique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Sciences et ingénierie de l'environnement	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue d'enseignement	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	**220**

Remarque

Cours en classe inversée, merci de consulter https://go.epfl.ch/classes-inversees avant de vous inscrire

Résumé

Le but du cours de physique générale est de donner à l'étudiant les notions de base nécessaires à la compréhension des phénomènes physiques. L'objectif est atteint lorsque l'étudiant est capable de prévoir quantitativement les conséquences de ces phénomènes avec des outils théoriques appropriés.

Contenu

Introduction et Cinématique : référentiels, trajectoire, vitesse, accélération, coordonnées cartésiennes cylindriques.
Dynamique du point matériel : quantité de mouvement, lois de Newton, forces fondamentales, empiriques et de liaison, mouvement oscillatoire, moment cinétique.
Travail, puissance, énergie : énergies cinétique, potentielle, mécanique, lois de conservation, mouvements gravitationnels, collisions.
Changement de référentiels : dynamique dans les référentiels non inertiels
Dynamique des systèmes : centre de masse, moment cinétique, énergie
Solide indéformable : moment cinétique, moment d'inertie, effets gyroscopiques

Compléments
L'enseignement peut contenir, mais pas exclusivement, les éléments suivants: coordonnées sphériques, relativité restreinte

Mots-clés

Physique générale, mécanique du point matériel, mécanique du solide, coordonnées, cinématique, relativité, énergie, travail

Compétences requises

Cours prérequis indicatifs

- Fortes compétences en niveau mathématique de la maturité Suisse, voir par exemple "http://www.math.ch/kanon/catalogue/"
- "Savoir-Faire en Maths - bien commencer ses études scientifiques", Y. Biollay, A. Chaabouni, J. Stubbe, PPUR, 2010

Concepts importants à maîtriser

Algèbre des vecteurs: Produits scalaires et vectoriel, projection des vecteurs. Rélations du triangle rectangle.
Resolution des équations linéaires avec 2 ou 3 inconnus.
Intégration/différentiation des fonctions et des vecteurs.
Conversion des unités physiques

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Elaborer un modèle physique d'un système mécanique
- Démontrer un savoir-faire dans la résolution de problèmes
- Reconnaitre les propres unités
- Juger les approximations employées
- Identifier les comportments qualitatifs prévus
- Estimer les ordres de grandeur
- Relier les notions de cours et les observations du monde quotidien
- Prendre en considération les chiffres significatifs
- Dériver les équations du mouvement

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Cours en classe inversée. demandant une grande indépendance de travail. Les vidéos du cours magistral doivent être vues et travaillées avant les séances en amphi avec l'enseignante.
Des exercices en séance avec des assistants complètent cet enseignement.

Travail attendu

Hors cours et exercices (ces indications peuvent varier selon votre préparation aux études ainsi que l'organisation du travail):
Avant le cours, visualiser les chapitre de la semaine sur switchtube, en prenant des notes de la manière qui vous convient, comme pour un cours en amphi classique.
Pendant les séances en amphi, participer activement (quizzes et exercices) et noter les points à reprendre du cours magistral.
Se préparer aux exercices avant les séances avec les assistant.e.s

Méthode d'évaluation

Examen écrit à la session d'hiver

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui
Autres	disponibilité autour de l'amphi. L'enseignante suit et participe au forum.

Ressources

Bibliographie

- Mécanique : fondements et applications, J.-Ph Pérez, Dunod
- Physique générale I: mécamique et thermodynamique; Alonso et Finn, Dunod

Ressources en bibliothèque

- Physique générale I / Alonso
- Mécanique / Pérez

Polycopiés

Poly très sythétique mis à disposition
Slides vierges et annotées des vidéos

Sites web

- http://lsme.epfl.ch

Liens Moodle

- https://go.epfl.ch/PHYS-101_I

Vidéos

- https://mediaspace.epfl.ch/channel/Physique+G%C3%A9n%C3%A9rale/28979

Préparation pour

Physique générale - Thermodynamique

CS-108 **Pratique de la programmation orientée-objet**

Schinz Michel

Cursus	Sem.	Type
Informatique	BA2	Obl.
Systèmes de communication	BA2	Obl.

Langue d'enseignement	français
Coefficient	9
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	270h
Semaines	14
Heures	**10 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Projet	6 hebdo
Nombre de places	

Résumé

Les étudiants perfectionnent leurs connaissances en Java et les mettent en pratique en réalisant un projet de taille conséquente. Ils apprennent à utiliser et à mettre en œuvre les principaux types de collections (listes, ensembles, tables associatives), et examinent quelques patrons de conception.

Contenu

Approfondissement des connaissances du langage Java, en particulier de la généricité (polymorphisme paramétrique), des classes imbriquées et anonymes et des lambdas.
Introduction à différents aspects de la bibliothèque standard Java : collections, entrées-sorties, interfaces utilisateur graphiques, etc.
Etude des mises en œuvre des collections par chaînage, arbres binaires de recherche ou hachage.
Introduction aux patrons de conception (*design patterns*) et examen des plus importants (*Decorator*, *Composite*, *Builder*, etc.).
Examen de l'utilisation judicieuse de l'héritage et de l'immuabilité.
Réalisation d'un projet de programmation conséquent en Java.

Mots-clés

Java, programmation orientée-objets, collections, patrons de conception.

Compétences requises

Cours prérequis obligatoires

CS-107 Introduction à la programmation.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Concevoir et écrire des programmes Java de taille moyenne.
- Utiliser à bon escient la totalité des concepts de Java.
- Utiliser et concevoir des classes et méthodes génériques en Java.
- Utiliser et mettre en oeuvre les principales sortes de collection (listes, ensembles, tables associatives).
- Utiliser judicieusement l'héritage et l'immuabilité dans les langages orienté-objets.
- Reconnaitre et savoir utiliser plusieurs patrons de conception.

Compétences transversales

• Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.

Méthode d'enseignement

Ex-cathedra.

Travail attendu

Participation au cours, réalisation des exercices, réalisation du projet.

Méthode d'évaluation

Durant le semestre : projet (60%), examen intermédiaire (15%) et examen final (25%).

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Oui

Ressources en bibliothèque

- Java Generics and Collections / Naftalin
- Effective Java / Bloch

Sites web

- https://cs108.epfl.ch/

MATH-310

Algebra

Lachowska Anna

Cursus	Sem.	Type
Chemistry	BA5	Opt.
Communication systems	BA5	Obl.
Computer science	BA3	Opt.
Cyber security minor	H	Opt.
HES - IC	H	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This is an introduction to modern algebra: groups, rings and fields.

Content

Integer numbers, Bezout's theorem. Groups, dihedral and symmetric groups. Group homomorphisms. Classification of finite abelian groups. Rings, ideals. Polynomial rings. Integral domains and Euclidean domains. Finite fields.

Keywords

Group, homomorphism, subgroup, normal subgroup, quotient group, cyclic group, symmetric group, order of the group, order of an element in the group, finite abelian groups. Ring, ideal, principal ideal, maximal ideal, principal ideal domain, Euler's totient function, field, finite field, characteristic of a field.

Learning Prerequisites

Required courses

Linear algebra

Recommended courses

Linear Algebra I, Analyse I, Analyse II

Learning Outcomes

By the end of the course, the student must be able to:

- Apply concepts and ideas of the course
- Reason rigorously using the notions of the course
- Choose an appropriate method to solve problems
- Identify the concepts relevant to each problem
- Apply concepts to solve problems similar to the examples shown in the course and in problem sets
- Solve new problems using the ideas of the course
- Implement appropriate methods to investigate the structure of a given group, ring or field, and study their properties
- Detect properties of algebraic objects
- Analyze finite groups
- Formulate structure of a finite abelian group in terms of cyclic groups
- Analyze structure of a ring, in particular polynomial rings

Teaching methods

Lectures and exercise sessions

Assessment methods

Written homework assignment (10% of the grade)
Written exam (90 % of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

1. D.S. Dummit, R. M. Foote, Abstract Algebra. Wiley, Third Edition
2. S. Lang, Undergraduate Algebra. Undergraduate texts in Mathematics. Springer-Verlag, Inc. New York, second edition, 1990.
3. L. Childs, A Concrete Introduction to Higher Algebra. Undergraduate texts in Mathematics, Springer-Verlag, Inc. New York, 1995.

Ressources en bibliothèque

- Undergraduate Algebra / Lang
- Abstract algebra /Dummit
- A Concrete Introduction to Higher Algebra / Childs

Notes/Handbook

Complete lecture notes will be available in PDF

Moodle Link

- https://go.epfl.ch/MATH-310

Videos

- https://mediaspace.epfl.ch/channel/MATH-310+Algebra/30044/subscribe

CS-250

Algorithms I

Chiesa Alessandro, Svensson Ola Nils Anders

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA4	Obl.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science minor	E	Opt.
Computer science	BA4	Obl.
Cyber security minor	E	Opt.
Data science minor	E	Opt.
HES - IC	E	Obl.
Mathematics	BA6	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

The students learn the theory and practice of basic concepts and techniques in algorithms. The course covers mathematical induction, techniques for analyzing algorithms, elementary data structures, major algorithmic paradigms such as dynamic programming, sorting and searching, and graph algorithms.

Content

Mathematical Induction

- Mathematical background, Euler's formula for trees.

Analysis of Algorithms

- O-notation, time and space complexity, recurrence relations, probabilistic analysis.

Data structures

- Arrays, linkes lists, trees, heaps, hashing, graphs.

Design of algorithms by induction

- Divide-and-conquer algorithms, dynamic programming.

Greedy Algorithms

- Spanning tree and shortest path algorithms.

Sorting and searching

- **merge sort, bucket sort, quicksort, heapsort, binary search.**

Graphs algorithms and data structures

- Graphs traversals, shortes path, spanning trees, transitive closures, decompositions, matching, network flows.

Keywords

Algoriths, data structures, efficiency, problem solving

Learning Prerequisites

Recommended courses

Learning Outcomes

By the end of the course, the student must be able to:

- Illustrate the execution of algorithms on example inputs
- Describe basic data structures such as arrays, lists, stacks, queues, binary, search trees, heapas, and hash tables
- Analyze algorithm efficiency
- Compare alternative algorithms and data structures with respect to efficiency
- Choose which algorithm or data structure to use in different scenarios
- Use algorithms and data structures taught in the course on concrete problem instances
- Design new algorithms and data structures bases on known methods
- Prove the corrrectness of an algorithm

Teaching methods

Ex cathedra lecture, exercises in classroom

Assessment methods

Continous assessment with final exam.

Resources

Moodle Link

- https://go.epfl.ch/CS-250

MATH-203(d)

Analysis III (for IC)

Licht Martin Werner

Cursus	Sem.	Type
Communication systems	BA3	Obl.
Computer science	BA3	Obl.
HES - IC	H	Obl.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly
Number of positions	

Summary

The course studies the fundamental concepts of vector analysis and Fourier analysis with a view to using them to solve multidisciplinary scientific engineering problems.

Content

Vector analysis:
Gradient, rotational, divergence and Laplacian operators. Curvilinear integrals and surface integrals. Vector fields and potentials. Green's, divergence and Stokes' theorems.
Fourier analysis:
Fourier series. Parceval identity. Fourier transforms. Plancherel's identity. Uses and applications.

Learning Prerequisites

Required courses

Analyse I, Analyse II, Algèbre linéaire.

Assessment methods

Written exam.

Resources

Moodle Link

- https://go.epfl.ch/MATH-203_d

MATH-207(d)

Analysis IV

Licht Martin Werner

Cursus	Sem.	Type
Chemistry	BA6	Opt.
Civil Engineering	BA4	Obl.
Communication systems	BA4	Opt.
Computer science	BA4	Opt.
Environmental Sciences and Engineering	BA4	Obl.
HES - SIE	E	Obl.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Resources

Moodle Link

- https://go.epfl.ch/MATH-207_d

EPFL

COM-304 **Communications project**

Al Hassanieh Haitham, Zamir Amir

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	8
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**12 weekly**
Lecture	2 weekly
Project	10 weekly
Number of positions	

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course teaches the development of systems that solve real-world challenges in communications, signal processing, AI, and robotics. Students will work in teams, construct their ideas, and either program available hardware prototypes or build their hardware.

Content

The course will teach students both technical and project management skills which are essential in developing, designing, and prototyping practical systems where the underlying challenges fall in on or multiple areas with a focus on communication, signal processing, AI, and Robotics.
The primary goal of this course is to give students hands-on experience with solving real-world challenges by working in teams to program different hardware platforms and ultimately build their own projects. The overall structure of the course will consist of a few introductory lectures at the beginning to introduce the project and research areas in wireless radar sensing, communication, computer vision, robotics, and reinforcement learning. The students will have time to go through the background material needed for the course and get familiar with the hardware and sensor platforms. Students will then organize into groups of 3 or 4 and propose their project using one or more of the provided hardware platforms, with the aid of the course staff. Finally, students will design and build their own project.
This class has two types of lectures.
(1) In person lectures which are limited to three lectures at the beginning of the semester. After which the lecture time will be used as office hours to help students with their projects.

1. Lecture 1: Class Introduction
2. Lecture 2: Introduction to Wireless Communications & Sensing
3. Lecture 3: Introduction to Reinforcement Learning & Robotics

(2)Online lectures on background material.

1. Wireless Communication
2. Radar Signal Processing
3. Reinforcement Learning

The class will support 3 hardware platforms which students can work with.

- Millimeter Wave TI AWR1443BOOST Radars
- BladeRF 2.0 micro xA4 Software Defined Radios
- Turtlebot 4 Lite

We have two additional robots which students can use. However, we do not provide support for these robots and we only have one available from each type.

- Unitree Go 2 EDU

- MyAGV

Learning Prerequisites

Recommended courses

COM-202 Signal Processing (BA3)
CS-233 Introduction to Machine Learning (BA4)
COM-302 Principles of Digital Communications (BA6) (To be taken concurrently)
CS-202 Computer System (BA4)
COM-102 Advanced Information, Computation, Communications II (BA2)

Important concepts to start the course

Basic programming skills.

Teaching methods

- Video lecture on background material.
- Tutorials on the hardware prototypes.
- Continuous supervision and tutoring
- Extensive team work and team feedback

Expected student activities

- Take an entrepreneurial approach to create and develop a practical system under the given hardware constraints.
- Work with team members to complete a large practical project
- Independently research solutions, learn new concepts and apply them in practice.
- Debug software/hardware systems.
- Discuss project progress in class
- Provide constructive criticism and feedback to other groups
- Present project outcome in a public forum

Assessment methods

35% Individual activities grade
65% Team project grade

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

CS-200

Computer architecture

Ienne Paolo

Cursus	Sem.	Type
Communication systems	BA3	Opt.
Computer science minor	H	Opt.
Computer science	BA3	Obl.
HES - IC	H	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course completes the overview of computer architecture started on the first year (CS-173 DSD).

Content

- Complex digital systems in Verilog
- Basic components of a computer
- Instruction Set Architectures (RISC-V)
- Memory Hierarchy
- IOs and Exceptions
- Instruction Level Parallelism
- Multiprocessors and Cache Coherence

Learning Prerequisites

Required courses

CS-173 Fundamentals of digital systems

Recommended courses

None

Important concepts to start the course

- Digital logic (combinational and sequential circuits, FSMs)
- Basic notions of processors and assembly

Learning Outcomes

By the end of the course, the student must be able to:

- Structure nontrivial assembly language programs
- Add interrupt handling logic in a processor and write simple exception handlers in assembler
- Understand the design principles of a modern memory hierarchy

- Understand the interaction mechanisms of system software with hardware
- Design pipelined digital circuits at Register Transfer Level
- Optimize the performance of a processor pipeline by reordering instructions

Teaching methods

- Ex cathedra
- Exercises
- Projects

Expected student activities

- Attending the course and exercise/lab sessions
- Completing the exercises and lab assignments
- Participating in the discussion on the forum

Assessment methods

- Graded lab assignments
- Midterm exam
- Written exam

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-200

Prerequisite for

Computer Systems

EPFL

CS-341 **Computer graphics**

Pauly Mark

Cursus	Sem.	Type
Communication systems	BA4	Opt.
Computer science minor	E	Opt.
Computer science	BA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

The students study and apply fundamental concepts and algorithms of computer graphics for rendering, geometry synthesis, and animation. They design and implement their own interactive graphics programs.

Content

This course provides an introduction to the field of Computer Graphics. We will cover elementary rendering algorithms such as rasterization and raytracing, examine mathematical concepts and algorithms for geometric modeling, and then study concepts and algorithms for computer animation. Students will experiment with modern graphics programming and build small interactive demos. Complemented by some theoretical exercises, these programming tasks lead to a graphics software project, where small teams of students design and implement a complete graphics application.

Keywords

Pixels and images, 2D and 3D transformations, perspective transformations and visibility, rasterization, interpolation and lighting, raytracing, shader programming, texture mapping, procedural modeling, curves and surfaces, polygonal meshes, particle systems

Learning Prerequisites

Required courses

Linear Algebra, Calculus

Recommended courses

CS-328 Numerical methods for visual computing

Learning Outcomes

By the end of the course, the student must be able to:

- Explain and apply the fundamental mathematical concepts of computer-based image and geometry synthesis
- Implement a basic rendering pipeline based on rasterization and raytracing
- Design and implement geometry synthesis based on procedural modeling
- Design and implement basic computer animation algorithms
- Integrate individual components into a complete graphics application
- Coordinate a team during a software project

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project. Exercises and project are done in groups of three students.

Assessment methods

- Programming homeworks and group project
- Final written examination

Resources

Notes/Handbook

Slides and online resources will be provided in class

Moodle Link

- https://go.epfl.ch/CS-341

Prerequisite for

Advanced Computer Graphics

CS-320

Computer language processing

Kuncak Viktor

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science minor	E	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

We teach the fundamental aspects of analyzing and interpreting computer languages, including the techniques to build compilers. You will build a working compiler from an elegant functional language into machine code using a popular backend called LLVM (https://llvm.org)

Content

See https://lara.epfl.ch/w/cc

1. Overview, source languages and run-time models
2. Review of formal languages
3. Lexical analysis
4. Syntactic analysis (parsing)
5. Name analysis
6. Type checking
7. Code generation
8. Correctness of compilers

Keywords

programming language;
compiler;
interpreter;
regular expression;
context-free grammar;
type system;
code generation;
static code analysis

Learning Prerequisites

Recommended courses

Discrete Mathematics
Theory of computation
Functional Programming
Computer architecture

Learning Outcomes

By the end of the course, the student must be able to:

- Design a programming language
- Construct a compiler
- Coordinate development with project partner
- Formulate correctness conditions for compiler
- Estimate time to implement a programming language feature
- Produce a working programming language implementation
- Decide which language features make implementation difficult
- Specify programming language and compiler functionality

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Respect the rules of the institution in which you are working.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate a capacity for creativity.
- Take feedback (critique) and respond in an appropriate manner.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Lectures, exercises, labs

Expected student activities

- Follow lectures
- Project work, indepdently and under supervision of assistants

Assessment methods

The grade is based on a midterm exam (30%) as well as programming, testing, documentation, and presentation of several projects done on student's own laptops during the semester (70%).
Different groups of students may be assigned different variants of projects. There may be small but unavoidable variations in the difficulty of different variants.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Andrew W. Appel, **Modern compiler implementation in Java** (or **ML**), Addison-Wesley 1997 (full PDF available from EPFL library)
Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: **Compilers: Principles, Techniques, and Tools** (2nd Edition, 2006)

Ressources en bibliothèque

- Modern compiler implementation in Java / Appel

- Compilers, principle, techniques and tools / Aho

Notes/Handbook

http://lara.epfl.ch/w/cc
Faboulous and gently paced videos: https://www.coursera.org/course/compilers

Websites

- https://lara.epfl.ch/w/cc

Moodle Link

- https://go.epfl.ch/CS-320

Prerequisite for

Advanced compiler construction
Recommended for Foundations of software

COM-301

Computer security and privacy

Bugnion Edouard, Troncoso Carmela

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Communication systems	BA5	Obl.
Computer science minor	H	Opt.
Computer science	BA5	Obl.
Cyber security minor	H	Opt.
HES - IC	H	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This is an introductory course to computer security and privacy. Its goal is to provide students with means to reason about security and privacy problems, and provide them with tools to confront them.

Content

The goal of this course is to introduce students to security engineering. The course will help students to think as an adversary so that they can analyse systems and establish security policies. We will cover a number of common security mechanisms at all layers, and learn their properties and limitations.
Core topics:

- Security design principles
- Access control
- Authentication mechanisms
- Applied cryptography
- Sotware and Network security
- Privacy

Keywords

Security Privacy

Learning Prerequisites

Recommended courses

CS-233a or CS-233b Introduction to Machine Learning (for programming)
COM-208 Computer Networks
CS-323 Introduction to operating systems

Important concepts to start the course

Basic notions TCP/IP
Basic notions programming

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze systems for security
- Decide on security mechanisms to apply
- Establish a security policy

Teaching methods

Pre-recorded lectures
Practical assignemnts interactivel resolved in class using the concepts learned in the lectures
Written exercises to reaffirm the learning of the course
Practical programming homeworks to develop attacks and defenses

Expected student activities

Attending lectures, solving exercises, reading and demonstrating understanding of provided materials.

Assessment methods

- Take home exams (80%)
- Practical homeworks (20%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Computer security by Dieter Gollmann

Security Engineering by Ross Anderson

Computer Security: Principles and Practice by Stallings and Brown

Ressources en bibliothèque

- Security Engineering / Anderson
- Computer security / Gollmann
- Computer Security / Stallings & Brown

Moodle Link

- https://go.epfl.ch/COM-301

EPFL

CS-202 **Computer systems**

Argyraki Katerina, Bugnion Edouard, Chappelier Jean-Cédric

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA4	Obl.
Computer science minor	E	Opt.
Computer science	BA4	Obl.
Cyber security minor	E	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course will teach operating systems and networks in an integrated fashion,emphasising the fundamental concepts and techniques that make their interaction possible/practical. Core lectures will be coupled with C programming lectures and assignments for hands-on experience.

Content

We will communicate these concepts and techniques through examples at different layers of the stack and draw connections/parallels between different aspects of computing systems.

- The role of the Operating System
- File systems
- Virtual memory
- Efficient resource management
- Networked applications
- The Internet
- Transport layer
- Network layer
- Link layer
- Data-centers and cloud systems

Learning Prerequisites

Required courses

- CS-107 Introduction à la programmation
- CS-108 Pratique de la programmation orientée-objet
- CS-173 Fundamentals of Digital Systems

Important concepts to start the course

- Basic Programming
- Basic computer architecture

Learning Outcomes

By the end of the course, the student must be able to:

- Manage key elements of operating systems and networks
- Critique the design of an OS or network protocol
- Design and implement C programs and network applications
- Compare different OS functions and network protocols
- Investigate simple OS and network attacks
- Investigate the correctness of C programs through debugging

Teaching methods

- Ex cathedra
- Hands-on exercise sessions

Expected student activities

- Participate in lectures and exercise sessions
- Answer quizzes
- Submit programming assignments
- Take midterm and final exams

Assessment methods

- Programming assignments (by groups of two)
- Midterm and final exam
- Quizzes

Supervision

Office hours	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-202

Prerequisite for

CS-311 The Software Enterprise - from ideas to products
CS-300 Data-Intensive Systems

CS-300 **Data-intensive systems**

Ailamaki Anastasia, Kashyap Sanidhya

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science minor	E	Opt.
Computer science	BA6	Obl.
Cyber security minor	E	Opt.
Data science minor	E	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

This course covers the data management system design concepts using a hands-on approach.

Content

We will cover topics such as database architecture, data models, query optimization and planning, query engine and system programming, and storage systems. As our field is evolving rapidly, the list below is not exhaustive and will be adapted to reflect current technology trends.

- Entity-relationship and relational model
- Relational Algebra, Calculus, and SQL
- Memory and storage hierarchy
- Sorting and Indexing
- Query operators and optimization
- Basic synchronization mechanisms: locking and latching, task coordination
- Transaction management and concurrency control

Learning Prerequisites

Required courses

CS-202 Computer Systems
CS-302 Parallelism and concurrency in software

Recommended courses

CS-200 Computer architecture

Important concepts to start the course

- Excellent C/C++ programming skills
- Undergraduate computer science course(s) on algorithms and data structures

Learning Outcomes

By the end of the course, the student must be able to:

- Identify and manage key components of database and operating systems
- Choose or critique design choices for DB and OS system software
- Express application information requirements and model the data of an application
- Create and design a database with a practical application in mind while justifying choices
- Explore how a DBMS works
- Report performance and possible optimizations for applications utilizing a DBMS

Teaching methods

Lectures, exercices, and projects

Expected student activities

- Attend the lectures to ask questions and interact with the professor
- Attend the exercises session to solve and discuss exercises about the recently taught material
- Complete a team project that covers the practical side of the course, e.g., build a set of key components in a DBMS
- Study all the material provided and recommended during classes
- Take a midterm and a final exam

Assessment methods

- Labs (30%)
- Midterm (25%)
- Final exam (45%)

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-300

EE-200

Electromagnétisme I : lignes et ondes

Fleury Romain

Cursus	Sem.	Type
Génie électrique et électronique	BA3	Obl.
HES - EL	H	Obl.
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue d'enseignement	français
Crédits	3
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo
Nombre de places	

Résumé

Le signal électrique est un vecteur essentiel pour la transmission d'information et d'énergie. En haute fréquence elle se manifeste comme un signal électromagnétique dont l'étude demande le développement de modèles physiques et mathématiques spécifiques basés sur les équations d'onde.

Contenu

1) Composants électroniques localisés ou distribués

1. Limites de la théorie des circuits localisés
2. Temps de montée et temps de propagation
3. Période et temps de propagation
4. Taille du composant et longueur d¿onde
5. Les différents types de composants électroniques

2) Théorie des lignes de transmission : domaine temporel

1. Discussion heuristique
2. Equations de base et solutions
3. Réflexions aux discontinuités
4. Terminaisons résistives
5. Terminaisons réactives
6. Terminaisons non linéaires : diagramme de Bergeron
7. Application : réflectométrie en domaine temporel
8. Paramètres des lignes de transmissions courantes

3) Théorie des lignes de transmission : domaine fréquentiel

1. Ondes monochromatiques et phaseurs complexes
2. Lignes terminées par un court-circuit ou un circuit ouvert
3. Lignes terminées par une impédance arbitraire
4. Flux de puissance sur une ligne de transmission
5. Adaptation d'impédance
6. Abaque de Smith
7. Effet des pertes et absorption
8. Systèmes à deux ports : paramètres S, Z, et M

Mots-clés

Signal électromagnétique, Circuits Distribués, Lignes de transmission, Ondes électromagnétiques Guidées, Réflexion et transmission, Circuits équivalents, Circuits radiofréquences, Impédance.

Compétences requises

Cours prérequis obligatoires

EE-100, Science et technologies de l'électricité

Cours prérequis indicatifs

Algèbre, Analyse I et II, Physique générale

Concepts importants à maîtriser

Critères de validité de l'hypothèse des circuits localisés.
Propriétés du signal électromagnétique: vitesse, fréquence, longueur d'onde.
Nature et comportement des signaux et ondes électromagnétiques: propagation guidée unidimensionnelle (lignes de transmission), en domaine temporel et fréquentiel.
Diagrammes des reflections multiples. Réflectométrie en domaine temporel.
Notion de phaseur complexe. Abaque de Smith, Adaptation d'impédance.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Analyser un système à lignes de transmission
- Calculer la réponse d'un circuit distribué
- Concevoir un système adapté en impédance
- Utiliser un Abaque de Smith

Méthode d'enseignement

Ex cathedra avec exercices en salle. Un support de cours est fourni sur Moodle, contenant l'essentiel du cours, ainsi que des cadres vides pour permettre à l'étudiant de prendre notes des démonstrations et exemples effectués avec le professeur. Les exercices, en relation directe avec le cours, peuvent nécessiter l'utilisation d'un ordinateur (MatLab ou Mathematica).

Travail attendu

Participation active au cours et aux séances d'exercices.

Méthode d'évaluation

Examen écrit.

Ressources

Bibliographie

1) Support de cours par R. Fleury, disponible sur Moodle.
2) Engineering electromagnetics and waves, U. Inan.

Ressources en bibliothèque

- Engineering Electromagnetics and waves. U. Inan

Liens Moodle

- https://go.epfl.ch/EE-200

Préparation pour

Electromagnétisme II: calcul des champs. Transmissions Hyperfréquences et Optiques, Télécommunications, Réseaux électriques, Rayonnement et Antennes, Propagation, Audio, cycle Master EPFL-SEL et EPFL-SC

EE-201

Electromagnétisme II : calcul des champs

Fleury Romain

Cursus	Sem.	Type
Génie électrique et électronique	BA4	Obl.
HES - EL	E	Obl.
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue d'enseignement	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo
Nombre de places	

Résumé

Ce cours traite de l'électromagnétisme dans le vide et dans les milieux continus. A partir des principes fondamentaux de l'électromagnétisme, on établit les méthodes de résolution des équations de Maxwell dans le vide et dans des milieux matériels complexes.

Contenu

1) Rappels d'analyse vectorielle

1. Champs vectoriels et champ scalaires
2. Systèmes de coordonnées
3. Gradient, divergence et rotationnel

2) Théorie du champ électromagnétique

1. Principes fondamentaux*: Équations de Maxwell, Conservation de la charge, Champs monochromatiques, Relations constitutives, Conditions aux limites*
2. Théorèmes fondamentaux*: Théorème de Poynting, Dualité électromagnétique, Unicité du champ, Réciprocité de Lorentz*

3) Ondes planes monochromatiques

1. Relation de dispersion
2. Polarisation
3. Conducteurs et effet de peau
4. Coefficients de Fresnel
5. Théorie des lignes de transmission

4) Rayonnement en espace libre

1. Solution exacte*: Potentiel vecteur et potentiel scalaire, Jauge de Lorentz, Fonction de Green, Dipôle infinitésimal*
2. Solution en champ lointain
3. Méthode des images
4. Principe d'équivalence de Huygens
5. Limite de diffraction

5) Milieux dispersifs (si le temps le permet)

1. Matériaux plasmoniques : modèle de Drude
2. Relations de Kramers-Kronig

Mots-clés

electromagnetisme, théorie du champ, distributions de charges et courants électriques, propagation des ondes électromagnétiques, rayonnement, champ lointain, milieux continus

Compétences requises

Cours prérequis obligatoires

Physique Générale (Electromagnétisme)

Cours prérequis indicatifs

Analyse I, II, III

Concepts importants à maîtriser

Charges et courant, Champ électromagnétique, Ondes électromagnétiques (longueur d'onde, fréquence, vitesse, impédance caractéristique, polarisation), Radiation, polarisation de la matière, dissipation.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Théoriser les principes fondamentaux de l'électromagnétisme
- Comparer les différentes propriétés électromagnétiques des matériaux
- Calculer les champs électriques et magnétiques rayonnés par une distribution de charge

Méthode d'enseignement

Ex cathedra avec exercices en salle.

Travail attendu

Participation active au cours et aux séances d'exercices.

Méthode d'évaluation

Examen écrit.

Ressources

Bibliographie

Support de cours par R. Fleury, disponible sur Moodle.

Polycopiés

Disponible sur Moodle.

Liens Moodle

- https://go.epfl.ch/EE-201

Vidéos

- https://mediaspace.epfl.ch/channel/channelid/29714

Préparation pour

Transmissions Hyperfréquences et Optiques, Photonique, Télécommunications, Orientation Communications mobiles, Rayonnement et Antennes, Propagation, Audio

EPFL

EE-202(b)

Electronique I

Sallese Jean-Michel, Zysman Eytan

Cursus	Sem.	Type
Informatique	BA3	Opt.
Systèmes de communication	BA3	Opt.

Langue d'enseignement	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo
Nombre de places	

Résumé

Découvrir le monde de l'électronique depuis les lois fondamentales des composants discrets linéaires et non linéaires. Les circuits obtenus avec des assemblages de composants nécessitent de nombreuses techniques de modélisation et d'analyse ainsi que des vérification exploitant un simulateur

Contenu

Cours

- Composants passifs linéaires
- Techniques de résolution de circuits linéaires
- Les diodes
- introduction aux transistors
- Techniques de modélisation des composants non linéaires
- Simulation électronique

Exercices

L'étudiant appliquera les nombreuses méthodes vues en cours pour résoudre des exercices pratiques qui pourront être vérifiés avec la simulation.

Mots-clés

Composants passifs, composants actifs, composants linéaires, composants non linéaires, diodes, transistors, modélisation, simulation, Lois de Kirchhoff, Thévenin-Norton, Superposition, impédances complexes, fonctions de transfert, Bode, concept d'amplification.

Compétences requises

Cours prérequis obligatoires

Cours d'analyse: équation différentielles du premier et second ordre, nombres complexes, résolution de système d'équations linéaires.

Cours prérequis indicatifs

Electricité de base: électrostatique, électrocinétique.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Analyser des circuits complexes
- Modéliser des composants non linéaires
- Modéliser des circuits complexes
- Raisonner à partir de méthode d'observation
- Dessiner des comportements temporels et fréquentiels
- Interpréter des signaux de natures diverses
- Utiliser les bonnes méthodes de résolution

Compétences transversales

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.

Méthode d'évaluation

Plusieurs quiz répartis sur le semestre et au moins un travail écrit.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

- **Principes d'électronique: cours et exercices corrigés. Albert Paul Malvino ; trad. de l'américain par Bernard Boittiaux ; Paris : Dunod, 2002**

Ressources en bibliothèque

- Principes d'électronique / Malvino

Polycopiés

- liste de sites approfondissant les notions vues en cours
- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.

Liens Moodle

- https://go.epfl.ch/EE-202_b

Préparation pour

Électronique II

EE-203(b)

Electronique II (pour IC)

Sallese Jean-Michel, Zysman Eytan

Cursus	Sem.	Type
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue d'enseignement	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Maîtriser des blocs fonctionnels nécessitant un plus haut niveau d'abstraction. Réalisation de fonctions électroniques de haut niveau exploitant les amplificateurs opérationnels.

Contenu

Cours

- Modèles d'amplificateurs
- Bande passante des amplificateurs
- Familles logiques
- l'amplificateur opérationnel en réaction négative
- l'amplificateur opérationnel en réaction positive
- les filtres actifs d'ordre N
- l'amplificateur opérationnel et ses imperfections
- les bascules
- Conversion AN et NA

Exercices

Comme en électronique I, l'étudiant appliquera de nombreuses méthodes vues en cours pour résoudre des exercices pratiques qui pourront être vérifiés avec la simulation

Mots-clés

Amplificateur, Modèle de quadripôle, polarisation, schéma petit signaux, Filtres, bande passante, puissante statique, puissance dynamique, Slew-rate, Tchebychev, Butterworth, Trigger de Schmitt, comparateur, intégrateur, différentiateur, monostable, bistable, astable, générateur de signaux, marge de bruit, Fan-In, Fan-Out, Puissance dissipée, tension d'offset.

Compétences requises

Cours prérequis indicatifs

Électronique I

Concepts importants à maîtriser

- Les quadripôles
- l'amplificateur opérationnel
- Caractéristique d'une famille logique
- Conversion numérique-analogique et analogique-numérique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Concevoir Concevoir des filtres actifs
- Concevoir Concevoir des circuits amplificateurs
- Comparer Savoir comparer les différentes familles logiques
- Analyser Analyser la bande passante d'une fonction électronique
- Concevoir Concevoir des fonctions électroniques de haut niveau à partir d'amplificateurs opérationnels
- Implémenter Implémenter la notion de temps
- Synthétiser Synthétiser des circuits logiques
- Concevoir Concevoir des interfaces pour des circuits logiques et analogiques

Méthode d'enseignement

- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.

Méthode d'évaluation

Projet à rendre avant le vacances de Noël et examination orale en janvier.
Le projet est divisé en trois parties:

- dépannage d'un circuit volontairement défaillant
- analyse d'un circuit complexe
- conception d'une fonction électronique de haut niveau

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

- Principes d'électronique: cours et exercices corrigés. Albert Paul Malvino ; trad. de l'américain par Bernard Boittiaux ; Paris : Dunod, 2002
- The Art of Electronics. Paul Horowitz et Winfield Hill; Cambridge University Press, 2015

Ressources en bibliothèque

- Principes d'électronique / Malvino

- The Art of Electronics / Horowitz, Hill

Polycopiés

- liste de sites approfondissant les notions vues en cours
- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.
- Développements en cours sur tablet

Liens Moodle

- https://go.epfl.ch/EE-203_b

CS-330 Intelligence artificielle

Faltings Boi

Cursus	Sem.	Type
Informatique	BA6	Opt.
Mineur en Data science	E	Opt.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA6	Opt.

Langue d'enseignement	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Introduction aux techniques de l'Intelligence Artificielle, complémentée par des exercices de programmation qui montrent les algorithmes et des exemples de leur application à des problèmes pratiques.

Contenu

Le cours comporte trois segments qui traitent les 3 différents formes d'inférence logique : déduction, abduction et induction :

1. Représentation de connaissances en logique de prédicats, algorithmes d'inférence
2. Systèmes experts
3. Raisonnement imprécis et incertain
4. Algorithmes de recherche
5. Satisfaction de Contraintes
6. Diagnostic et Planification
7. Apprentissage supervisé
8. Apprentissage non-supervisé
9. Apprentissage bio-inspiré

Compétences requises

Cours prérequis indicatifs

CS-214 Software construction (ou ancien cours Functional programming)

Concepts importants à maîtriser

Logique de prédicats
Algorithmes de base
Théorie de probabilités
Programmation

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Choisir le bon type d'inférence pour une application
- Choisir la méthode la plus appropriée pour un certain type d'inférence
- Evaluer la faisabilité d'une application de l'Intelligence Artificielle

- Choisir, implémenter et décrire des algorithmes d'inférence déductive sur la base de calcul de prédicats
- Formuler des connaissances utilisant la logique des prédicats
- Décrire des méthodes d'inférence avec des informations imprécises et incertaines
- Choisir, implémenter et décrire des algorithmes de recherche et de satisfaction de contraintes
- Choisir et décrire des méthodes pour le diagnostic
- Choisir, implémenter et décrire des méthodes pour la planification
- Choisir, implémenter et décrire des méthodes d'apprentissage supervisé sur la base d'exemples
- Choisir, implémenter et décrire des méthodes d'apprentissage non-supervisé

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur

Travail attendu

Participation au cours et exercices: 4 heures/semaine
Lecture: 2 heures/semaine
Travail independant: 3 heures/semaine

Méthode d'évaluation

Test intermédiaire 30%, examen final 70%

Ressources

Bibliographie

Boi Faltings, Michael Schumacher : Intelligence Artificielle par la pratique, PPUR
(Russel & Norvig : Artificial Intelligence : A Modern Approach / Prentice Hall)

Ressources en bibliothèque

- Intelligence Artificielle par la pratique / Faltings
- Artificial Intelligence / Russell

Liens Moodle

- https://go.epfl.ch/CS-330

Préparation pour

Intelligent Agents

EPFL

CS-213 Interaction personne-système

Dillenbourg Pierre

Cursus	Sem.	Type
Informatique	BA4	Opt.
Learning Sciences		Opt.
Systèmes de communication	BA4	Opt.

Langue d'enseignement	français
Crédits	5
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	150h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Projet	2 hebdo
Nombre de places	

Résumé

La discipline de l'Interaction Homme-Machine (ou HCI : Human-Computer Interaction) vise à systématiquement placer le facteur humain dans la conception de systèmes interactifs.

Contenu

Introduction du cours

1. Introduction: Interaction Homme-Machine & Expérience Utilisateur (UX)

Phases projet

1. Recueillir les besoins de l'organisation et analyser la concurrence : Voix du Business & Voix du Marché
2. Comprendre les utilisateurs cibles et leurs besoins : Voix du Client
3. Spécifier les utilisateurs cibles et leurs contextes d'utilisation : Personas, Parcours utilisateurs & Vision UX
4. Structurer et nommer les contenus : Architecture de l'information
5. Concevoir la solution : Maquettage & Prototypage
6. Évaluer la solution : Tests utilisateurs

Focus thématiques

1. Charge cognitive
2. Data vizualisation
3. Accessibilité Web
4. Interaction styles
5. Deceptive Patterns

Conclusions

1. Résumé des concepts-clés & revue des examens précédents

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Détecter les défauts d'une interface
- Expliquer les défauts du point de vue de l'utilisateur
- Suggérer des améliorations à une interface
- Réaliser un test de "usability"
- Concevoir une expérience
- Analyser les données d'un utilisateur
- Recommander un style d'interaction pour une tâche donnée
- Planifier les différentes phases d'une démarche de conception centrée utilisateurs
- Définir une problématique UX
- Identifier des bonnes pratiques de design
- Déterminer les défauts d'une interface

- Expliquer les défauts du point de vue de l'utilisateur
- Proposer des recommandations d'optimisations d'une interface
- Analyser des données qualitatives et quantitatives
- Créer des personas et des parcours utilisateurs
- Concevoir des interfaces ergonomiques de façon itérative
- Réaliser un test utilisateur
- Prendre en considération les critères d'accessibilité web

Méthode d'enseignement

Cours ex-cathedra incluant des exemples proches de la pratique
Projet de groupe permettant aux étudiants de mettre en pratique les concepts du cours au travers de la conception d'une solution centrée sur les besoins et attentes des utilisateurs

Travail attendu

Créer un prototype interactif illustrant les principaux écrans et parcours utilisateurs du système à optimiser

Méthode d'évaluation

Projet (50%)
Examen écrit (50%)

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Liens Moodle

- https://go.epfl.ch/CS-213

Préparation pour

CS-486 Interaction design

COM-308

Internet analytics

Grossglauser Matthias

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA6	Opt.
Computer science	BA6	Opt.
Data science minor	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

Internet analytics is the collection, modeling, and analysis of user data in large-scale online services, such as social networking, e-commerce, search, and advertisement. This class explores a number of the key functions of such online services that have become ubiquitous over the past decade.

Content

The class seeks a balance between foundational but relatively basic material in algorithms, statistics, graph theory and related fields, with real-world applications inspired by the current practice of internet and cloud services.
Specifically, we look at social & information networks, recommender systems, clustering and community detection, search/retrieval/topic models, dimensionality reduction, stream computing, and online ad auctions. Together, these provide a good coverage of the main uses for data mining and analytics applications in social networking, e-commerce, social media, etc.
The course is combination of theoretical materials and weekly laboratory sessions, where we explore several large-scale datasets from the real world. For this, you will work with a dedicated infrastructure based on Hadoop & Apache Spark.

Keywords

data mining; machine learning; social networking; map-reduce; hadoop; recommender systems; clustering; community detection; topic models; information retrieval; stream computing; ad auctions

Learning Prerequisites

Required courses

COM-300 Modèles stochastiques pour les communications

Recommended courses

Basic linear algebra
Algorithms & data structures

Important concepts to start the course

Graphs; linear algebra; Markov chains; Python

Learning Outcomes

By the end of the course, the student must be able to:

- Explore real-world data from online services
- Develop framework and models for typical data mining problems in online services
- Analyze the efficiency and effectiveness of these models
- Data-mining and machine learning techniques to concrete real-world problems

Expected student activities

Lectures with associated homeworks explore the basic models and fundamental concepts. The labs are designed to explore very practical questions based on a number of large-scale real-world datasets we have curated for the class. The labs draw on knowledge acquired in the lectures, but are hands-on and self-contained.

Assessment methods

Project 35%, final exam 65%

Resources

Bibliography

C. Bishop, Pattern Recognition and MachineLearning, Springer, 2006
A. Rajaraman, J. D. Ullman: Mining of Massive Datasets, 2012
M. Chiang: Networked Life, Cambridge, Cambridge, 2012
D. Easley, J. Kleinberg: Networks, Crowds, and Markets, Cambridge, 2010
Ch. D. Manning, P. Raghavan, H. Schütze: Introduction to Information Retrieval, Cambridge, 2008

Ressources en bibliothèque

- Networks, Crowds, and Markets / Easley
- Pattern Recognition and Machine Learning / Bishop
- Introduction to Information Retrieval / Manning
- Networked Life / Chiang
- Mining of Massive Dataset / Leskovec

Moodle Link

- https://go.epfl.ch/COM-308

BIO-109 Introduction aux sciences du vivant (pour IC)

Zufferey Romain

Cursus	Sem.	Type
Informatique	BA4	Opt.
Systèmes de communication	BA4	Opt.

Langue d'enseignement	français
Crédits	6
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Ce cours présente les principes fondamentaux à l'œuvre dans les organismes vivants. Autant que possible, l'accent est mis sur les contributions de l'Informatique aux progrès des Sciences de la Vie.

Contenu

Le cours aborde la plupart des concepts fondamentaux des Sciences de la Vie.
Les sujets développés parce qu'ils sont à l'interface avec l'informatique incluent :
- alignement des séquences, assemblage de séquences en génome
- matrice de distances et déduction d'un arbre phylogénétique
- détection de domaines transmembranaires et de signaux de localisation subcellulaire dans une séquence d'acides aminés.

Mots-clés

Bioinformatique, génome, séquençage, évolution, communication intercellulaires

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Identifier les principales structures cellulaires et comprendre les méthodes utilisées pour les observer
- Identifier les segments informatifs d'un génome
- Appliquer des algorithmes pour résoudre des questions e
- Expliquer le processus de l'expression génique
- Analyser des données expérimentales brutes et en tirer des conclusions sensées

Compétences transversales

- Accéder aux sources d'informations appropriées et les évaluer.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.

Travail attendu

En plus de la participations active aux cours et aux exercices, 4 heures de travail personnel sont attendues.

Méthode d'évaluation

Examen écrit durant la session d'été.

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Ressources

Polycopiés

Les dias du cours et les séries d'exercices sont mises à disposition du Moodle.

Liens Moodle

- https://go.epfl.ch/BIO-109

CS-233

Introduction to machine learning

Fua Pascal, Salzmann Mathieu

Cursus	Sem.	Type
Communication systems	BA4	Obl.
Computer science	BA4	Opt.
Environmental Sciences and Engineering	BA6	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

Machine learning and data analysis are becoming increasingly central in many sciences and applications. In this course, fundamental principles and methods of machine learning will be introduced, analyzed and practically implemented.

Content

- Introduction: General concepts, data representation, basic optimization.
- Linear methods: Linear regression, least-square classification, logistic regression, linear SVMs.
- Nonlinear methods: Polynomial regression, kernel methods, K nearest neighbors
- Deep learning: Multi-layer perceptron, CNNs.

- Unsupervised learning: Dimensionality reduction, clustering.

Learning Prerequisites

Required courses

Linear Algebra

Important concepts to start the course

- Basic linear algebra (matrix/vector multiplications, systems of linear equations, SVD)
- Multivariate calculus (derivatives w.r.t. vector and matrix variables)
- Basic programming skills (labs will use Python).

Learning Outcomes

By the end of the course, the student must be able to:

- Define the following basic machine learning problems : regression, classification, clustering, dimensionality reduction
- Explain the main differences between them
- Derive the formulation of these machine learning models
- Assess / Evaluate the main trade-offs such as overfitting, and computational cost vs accuracy
- Implement machine learning methods on real-wolrd problems, and rigorously evaluate their performance using

cross-validation

Teaching methods

- Lectures
- Pen-and-paper exercise sessions
- Python lab with a mini project in groups of 3 students

Expected student activities

- Attend lectures
- Attend lab sessions
- Work on the weekly theory and coding exercises

Assessment methods

- Self-assessment via the solutions of the pen-and-paper exercises and coding labs
- Two milestones for the mini-proejct (10% of the grade each)
- Final exam (80% of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-233

EPFL

CS-308

Introduction to quantum computation

Lévêque Olivier, Urbanke Rüdiger

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science	BA6	Opt.
Minor in Quantum Science and Engineering	E	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Summary

The course introduces the paradigm of quantum computation in an axiomatic way. We introduce the notion of quantum bit, gates, circuits and we treat the most important quantum algorithms. We also touch upon error correcting codes. This course is independent of COM-309.

Content

Introduction to quantum computation
- Classical circuit model, reversible computation.
- Quantum bits, Hilbert space of N qubits, unitary transformations, measurement postulate.
- Quantum circuit model, universal sets of gates.
- Deutsch and Josza's problem and algorithm.

Basic algorithms
- Hidden sub-group problem and Simon's algorithm
- Mathematical parenthesis: factoring integers and period of arithmetic functions. Notions on continued fraction expansions.
- Quantum Fourier transform and the period finding algorithm
- Shor's factoring algorithm.
- Grover's search algorithm.

Error correcting codes
- Models of noise and errors.
- Shor and Steane error correcting codes.
- Stabilizer codes.
- Calderbank-Shor-Steane construction.

Keywords

Quantum computation, quantum circuits, universal gates, quantum Fourier transform, Deutsch-Josza's algorithm. Simon algorithm, Shor's algorithm, Grover's algorithm, entanglement, quantum error correction.

Learning Prerequisites

Required courses

Linear algebra course, basic probability course

Important concepts to start the course

Matrices, unitary matrices, eigenvectors, eigenvalues, inner product, algebra of complex numbers

Learning Outcomes

By the end of the course, the student must be able to:
- Explain the concept of quantum algorithm on the circuit model
- Describe universal gates
- Describe basic quantum algorithms
- Compute the evolution of a state through a circuit
- Apply the measurement postulate
- Manipulate algebraic expressions involving the Dirac notation
- Carry out implementation on public NISQ devices
- Give an example of an error correcting code

Teaching methods

Ex cathedra classes. Exercices. Use of IBM Q NISQ devices

Expected student activities

Participation in class, exercise sessions, use of IBM Q NISQ devices

Assessment methods

- Mini project on IBM Q experience
- Graded homeworks
- Written final exam

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Assistants answer questions during exercise sessions

Resources

Bibliography

N. David Mermin. Quantum Computer Science. An Introduction. Cambridge University Press.
Nielsen and Chuang. Quantum Computation and Information. Cambridge University Press.

Ressources en bibliothèque

- Quantum Computer Science / Mermin
- Quantum Computation and Information / Nielsen

Moodle Link

- https://go.epfl.ch/CS-308

COM-309

Introduction to quantum information processing

Macris Nicolas

Cursus	Sem.	Type
Communication systems	BA5	Opt.
Computer science	BA5	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Summary

Information is processed in physical devices. In the quantum regime the concept of classical bit is replaced by the quantum bit. We introduce quantum principles, and then quantum communications, key distribution, quantum entropy, and spin dynamics. No prior knowledge of quantum physics is required.

Content

Introduction a la mecanique quantique des systemes discrets.
- Polarization of photons, basic experiments
- Notion of quantum state, notion of measurement
- Quantum principles, notion of quantum bits, entanglement, no-cloning
- Bloch sphere

Cryptographie, Communications et Corrélations
- Secret key generation: BB1984 and B92 protocols
- Entanglement: EPR pairs
- Bell/CSCH inequalityl. Ekert protocol for a secret key generation
- Teleportaion, dense coding, distillation.

Spin and its dynamics
- Stern-Gerlach experiment, spin 1/2
- Dynamics of spin in magnetic fields, Rabi oscillations
- Manipulations of the spin and elementary quantum gates
- Introduction to the Jaynes-Cummings Model

Density matrices and Von Neumann entropy
- mixed states and entropy
- bipartite systems and entanglement entropy
- non-signalling and teleportaion revisited

Keywords

Polarization, spin, measurement, quantum bit, entanglement, key distribution, teleportaion, dense coding, Von Neumann entropy, spin dynamics.

Learning Prerequisites

Required courses

Linear algebra, basic probability

Important concepts to start the course

Vectors, matrices, eigenvalues, eigenvectors, projectors, inner product, algebraic manipulation of complex numbers, discrete probability distribution.

Learning Outcomes

By the end of the course, the student must be able to:

- Describe principles of quantum physics
- Illustrate quantum bits with photon polarization and spin
- Explain basic communication protocols like key distribution, dense coding, teleportation
- Describe how to manipulate qubits with magnetic fields
- Define quantum entropies and list basic properties
- Use Qiskit and/or Pennylane and NISQ devices

Teaching methods

Ex cathedra lectures, exercise session, practical implementationns typically with IBM Q machines.

Expected student activities

Participation in class, homeworks, hands-on exercises on IBM-Q.

Assessment methods

- miniprojet
- Graded homeworks
- Final written exam

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Assistants are in exercise session

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

David Mermin, *Quantum computer science, An introduction, Cambridge university press 2000.* Written for computer science students with no knowledge of physics.
Michel Le Bellac, *A short introduction to quantum information and quantum computation, Cambridge University Press*. A pedagogic book with an elementary introduction to the physics of the subject.
Neil Gershenfeld. *The physics of information technology. Cambridge University Press.* On basic information technologies useful in computer science, classical communications and quantum aspects.

Ressources en bibliothèque

- Quantum computer science / Mermin
- The physics of information technology / Gershenfeld
- A short introduction to quantum information and quantum computation / Le Bellac

Notes/Handbook

Yes, on web site

Prerequisite for

Introduction to quantum information processing
Classes in Quantum Science and Engineering

EPFL

CS-358(a) **Making intelligent things A**

Koch Christoph

Cursus	Sem.	Type
Communication systems	BA5	Opt.
Computer science	BA5	Opt.

Language of teaching	English
Credits	8
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**13 weekly**
Exercises	1 weekly
Project	12 weekly
Number of positions	**50**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course aims at teaching the prototyping of intelligent physical artifacts. It aims to solve real-world challenges by a combination of microcontroller programming, electronics, and computer -aided design and manufacturing. Student teams choose their own challeng in consultation with the teachers.

Content

The course will teach students essential skills in designing and prototyping intelligent physical artifacts, including microcontroller (such as Arduino and Raspberry PI) programming, practical electronics, and computer-aided design and manufacturing, using modern prototyping methods such as 3D printing and CNC milling.
The course will leverage and refine students' skillsets in computational thinking and in building advanced software artifacts, and aims to open new horizons for them by allowing them to explore new ways of connecting the learning outcomes of other IC courses with the physical world. A substantial emphasis will be put on engineering low-level (microcontroller-based) systems software.
The course will be structured into three phases - a first consisting of tutorials and crash courses on essential skill sets such as practical electronics and 3d printing; a second in which students individually build a precisely specified small intelligent thing under close guidance by the teaching staff; and a third - the main project phase - in which teams of students propose, design, and implement their own project.
Students will have access to a workshops and digital fabrication technologies such as laser cutters, CNC milling machines, and 3D printers through EPFL's Discovery Learning Labs. We will define a suitable format allowing all student teams to exchange insights and present progress throughout the semester; at the end of the semester there will be a public event to showcase the results of the projects.

Learning Prerequisites

Recommended courses

- CS-101 Advanced ICC I
- CS-173 Fundamentals of digital systems

Important concepts to start the course

Basic programming skills.

Learning Outcomes

By the end of the course, the student must be able to:

- Apply a design thinking methodology in a project of inventing and prototyping an intelligent thing
- Design and develop simple microcontroller-based electronic circuits with sensors and actuators
- Prove constructive feedback on other groups'projects
- Evaluate how to best integrate computational methods and digital fabrication tools to achieve project goals
- Design a suitable format and material for public presentation of project outcomes
- Assess / Evaluate own project progress and device adaptations of the project plan if necessary

Teaching methods

- (Video) lectures on background technology
- Hands-on tutorials on digital fabrication technologies in collaboration with the DLL
- Tutoring throughout the project
- Regular project critiques in a weekly forum - students will be encouraged to give each other feedback in addition to teachers' feedback.

Expected student activities

- Take an entrepreneurial approach to create and develop a new idea under physical constraints such as the feasibility and cost of fabrication
- Coordinate a project team and engage in collaborative problem solving
- Build basic microcontroller-driven electronic circuits with sensors and actuators
- Deal with resource constraints prevalent in microcontroller programming
- Program sensors and actuators; implement low-level timed protocols, such as pulse-width modulation
- Fabricate and evaluate prototypes using 3d printing and related technologies
- Discuss project progress in class
- Provide constructive criticism and feedback to other groups
- Present project outcome in a public forum

Assessment methods

- **20% individual project grade (phase 2)**
- **60% Team project grade (phase 3)**
- **20% Course participation / Critiques**

Supervision

Office hours	Yes
Assistants	Yes
Others	COUNSELLING BY DISCOVERY LEARNING LABS STAFF & AFFILIATE COUNSELLORS

Resources

Moodle Link

- https://go.epfl.ch/CS-358_a

CS-358(b) **Making intelligent things B**

Koch Christoph

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	8
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**13 weekly**
Exercises	1 weekly
Project	12 weekly
Number of positions	**75**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course aims at teaching the prototyping of intelligent physical artifacts. It aims to solve real-world challenges by a combination of microcontroller programming, electronics, and computer -aided design and manufacturing. Student teams choose their own challeng in consultation with the teachers.

Content

The course will teach students essential skills in designing and prototyping intelligent physical artifacts, including microcontroller (such as Arduino and Raspberry PI) programming, practical electronics, and computer-aided design and manufacturing, using modern prototyping methods such as 3D printing and CNC milling.
The course will leverage and refine students' skillsets in computational thinking and in building advanced software artifacts, and aims to open new horizons for them by allowing them to explore new ways of connecting the learning outcomes of other IC courses with the physical world. A substantial emphasis will be put on engineering low-level (microcontroller-based) systems software.
The course will be structured into three phases - a first consisting of tutorials and crash courses on essential skill sets such as practical electronics and 3d printing; a second in which students individually build a precisely specified small intelligent thing under close guidance by the teaching staff; and a third - the main project phase - in which teams of students propose, design, and implement their own project.
Students will have access to a workshops and digital fabrication technologies such as laser cutters, CNC milling machines, and 3D printers through EPFL's Discovery Learning Labs. We will define a suitable format allowing all student teams to exchange insights and present progress throughout the semester; at the end of the semester there will be a public event to showcase the results of the projects.

Learning Prerequisites

Recommended courses

- CS-101 Advanced ICC I
- CS-173 Fundamentals of digital systems

Learning Outcomes

By the end of the course, the student must be able to:

- Apply a design thinking methodology in a project of inventing and prototyping an intelligent thing

• Design and develop simple microcontroller-based electronic circuits with sensors and actuators
• Prove constructive feedback on other groups' projects
• Evaluate how to best integrate computational methods and digital fabrication tools to achieve project goals
• Design a suitable format and material for public presentation of project outcomes
• Assess / Evaluate own project progress and device adaptations of the project plan if necessary

Teaching methods

• (Video) lectures on background technology
• Hands-on tutorials on digital fabrication technologies in collaboration with the DLL
• Tutoring throughout the project
• Regular project critiques in a weekly forum - students will be encouraged to give each other feedback in addition to teachers' feedback.

Expected student activities

• Take an entrepreneurial approach to create and develop a new idea under physical constraints such as the feasibility and cost of fabrication
• Coordinate a project team and engage in collaborative problem solving
• Build basic microcontroller-driven electronic circuits with sensors and actuators
• Deal with resource constraints prevalent in microcontroller programming
• Program sensors and actuators; implement low-level timed protocols, such as pulse-width modulation
• Fabricate and evaluate prototypes using 3d printing and related technologies
• Discuss project progress in class
• Provide constructive criticism and feedback to other groups
• Present project outcome in a public forum

Assessment methods

• 20% individual project grade (phase 2)
• 60% Team project grade (phase 3)
• 20% Course participation / Critiques

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	COUNSELLING BY DISCOVERY LEARNING LABS STAFF & AFFILIATE COUNSELLORS

Resources

Moodle Link

• https://go.epfl.ch/CS-358_b

EPFL

COM-300 **Modèles stochastiques pour les communications**

Thiran Patrick

Cursus	Sem.	Type
HES - IC	H	Opt.
Informatique	BA5	Opt.
Mineur en Data science	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA5	Obl.

Langue d'enseignement	français
Crédits	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	4 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

L'objectif de ce cours est la maitrise des outils des processus stochastiques utiles pour un ingénieur travaillant dans les domaines des systèmes de communication, de la science des données et de l'informatique.

Contenu

- Rappels de probabilité: axiomes de probabilité, variable aléatoire et vecteur aléatoire.
- Quelques inégalités utiles.
- Processus stochastiques à temps continu et à temps discret : analyse du second ordre (stationarité, ergodisme, densité spectrale, relations de Wiener-Khintchine, réponse d'un système linéaire invariant à des entrées aléatoires, processus gaussien, processus ARMA, filtres de Wiener). Exemples d'application à des cas simples de détection optimale ou de traitement d'image.
- Processus de Poisson et bruit impulsif de Poisson. Exemple d'application aux transmissions sur fibres optiques.
- Chaînes de Markov à temps discret. Classification des états, chaînes ergodiques: comportement asymptotique, chaînes absorbantes: temps d'atteinte, marches aléatoires simples, processus de branchement. Exemples d'application à l'analyse d'un algorithme ou d'un système informatique distribué.
- Chaînes de Markov à temps continu. Classification des états, chaînes ergodiques: comportement asymptotique. Processus de naissance et de mort à l'état transitoire et stationnaire. Exemples d'application à l'analyse de files d'attente simples: définition, loi de Little, files M/M/1... M/M/s/K, M/G/1.

Mots-clés

Probabilité, Processus stochastique, Moments, stationarité, Processus gaussien, Processus de Poisson, Chaîne de Markov, File d'attente.

Compétences requises

Cours prérequis obligatoires

- Algèbre linéaire (MATH 111 ou équivalent).
- Analyse I, II, III (MATH 101, 106, 203 ou équivalent).
- Premier cours de probabilité (MATH 232 ou équivalent).
- Circuits et systèmes II (EE 205 ou équivalent), ou Signaux et systèmes (MICRO310/311 ou équivalent), pour les notions de base de théorie des systèmes (déterministes) linéaires.

Cours prérequis indicatifs

- Circuits et systèmes I (EE 204 ou équivalent) pour les notions de base de théorie des circuits.
- Analyse IV (MATH 204 ou équivalent) pour les notions d'analyse complexe.

Concepts importants à maîtriser

Notions d'algèbre linéaire, en particulier opérations matricielles (inversion, diagonalisation, vaelurs propres d'une matrice).
Notions d'analyse (fonctions d'une ou plusieurs variables réelles, suites et séries, équations différentielles ordinaires linéaires).
Notions de théorie des systèmes linéaires (convolution, transformées de Fourier, Laplace et en z).

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Modéliser un système aléatoire.
- Analyser un problème avec une composante aléatoire.
- Evaluer les solutions d'un problème avec une composante aléatoire.

Méthode d'enseignement

- Ex cathedra (au tableau), 4h par semaine.
- Séances d'exercices, 2h par semaine.

Travail attendu

- Exercices en séance et à domicile

Méthode d'évaluation

- Examens intermédiaires pendant le semestre.
- Examen final en session.

Ressources

Bibliographie

Polycopié; textes de référence sur la page moodle du cours.

Ressources en bibliothèque

- Markov Chains / Norris
- Stochastic Processes / Ross
- Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues / Brémaud
- Probability and Random Processes / Grimmett & Stirzaker
- Introduction to Probability Models / Ross (10th ed.)
- Probability, Random Variables, and Stochastic Processes / Papoulis (4th ed.)

Polycopiés

Polycopié disponible sur la page moodle du cours.

Liens Moodle

- https://go.epfl.ch/COM-300

Préparation pour

Tous les cours en systèmes de communication, science des données et informatique utilisant des modèles stochastiques ou des méthodes aléatoires.

PHYS-202

Mécanique analytique (pour SPH)

Jacquod Philippe René Joseph

Cursus	Sem.	Type
Biologie computationnelle et quantitative		Opt.
Informatique	BA3	Opt.
Physique	BA3	Obl.
Systèmes de communication	BA3	Opt.

Langue d'enseignement	français
Crédits	5
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	3 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Présentation des méthodes de la mécanique analytique (équations de Lagrange et de Hamilton) et introduction aux notions de modes normaux et de stabilité.

Contenu

1. Rappels de mécanique newtonienne
2. Les équations de Lagrange- Principe de d'Alembert.- Principe de moindre action.- Coordonnées normales.
3. Les équations de Hamilton- Crochets de Poisson.- Transformations canoniques.- Méthode de Hamilton-Jacobi.
4. Introduction aux systèmes dynamiques- Notion de stabilité.- Modes Normaux.

Compétences requises

Cours prérequis indicatifs

Physique générale, Analyse, Algèbre linéaire

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Résoudre un problème en mécanique

Compétences transversales

- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.

Méthode d'enseignement

Ex cathedra et exercices en salle.

Méthode d'évaluation

examen écrit

Ressources

Bibliographie

Polycopié. "Classical Mechanics", H. Goldstein

Ressources en bibliothèque

- Classical Mechanics / Goldstein

Liens Moodle

- https://go.epfl.ch/PHYS-202

Préparation pour

Mécanique statistique, Physique quantique

BIOENG-310

Neuroscience foundations for engineers

Schrimpf Martin, Zenk Fides

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science	BA6	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	3 weekly
Number of positions	

Summary

This overview course bridges computational expertise with neuroscience fundamentals, aimed at fostering interdisciplinary communication and collaboration for engineering-based neuroscience programs.

Content

All content will combine experimental data and findings with computational models.

- Introduction to cellular and molecular biology
- Introduction to neuroscience
- Neural communication
- Individual neurons and small neural populations
- Sensory systems and perception I - Vision
- Sensory systems and perception II -Audition, Somatosensation
- Learning: Neuroplasticity and Molecular Mechanisms of Learning
- Motor Control I - Systems
- Motor Control II - Molecular
- Cognitive Neuroscience I - Sytems
- Cognitive Neuroscience II - Molecular
- Molecular Genetics in Neuroscience
- Neuropharmacology and Drug Design

Learning Prerequisites

Important concepts to start the course

Programming

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret experimental data in neuroscience
- Analyze experimental data in neuroscience
- Describe basic concepts in biology, neuroscience
- Describe basic computational tools and models in neuroscience

- Integrate biological and computational concepts in neuroscience

Teaching methods

- Lectures
- Practical tutorials and exercises
- Journal Club or Poster Presentation
- Excursions

Expected student activities

- Attend lectures and take notes
- Participate and prepare for tutorials and exercises
- Work on exercises in a group

Assessment methods

- 70% final exam
- 15% computational exercise
- 15% presentation exercise

Supervision

Office hours	No
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/BIOENG-310

Prerequisite for

This course will prepare students for the Neuro-X Master program.

CS-328

Numerical methods for visual computing and ML

Cursus	Sem.	Type
Communication systems	BA3	Opt.
Computer science	BA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

Visual computing and machine learning are characterized by their reliance on numerical algorithms to process large amounts of information such as images, shapes, and 3D volumes. This course will familiarize students with a range of essential numerical tools to solve practical problems in this area.

Content

This course provides a first introduction to the field of numerical analysis with a strong focus on visual computing and machine learning applications. Using examples from computer graphics, deep neural networks, geometry processing, computer vision, and computational photography, students will gain hands-on experience with a range of essential numerical algorithms.

The course will begin with a review of floating point arithmetic and error propagation in numerical computations. Following this, we will study and experiment with several techniques that solve systems of linear and non-linear equations and perform dimensionality reduction. Since many interesting problems cannot be solved exactly, numerical optimization techniques constitute the second major topic of this course. We will take an extensive look at automatic differentiation, the mechanism underlying popular deep learning frameworks such as PyTorch and Tensorflow. The course concludes with a review of numerical methods that introduce randomness to solve problems that would otherwise be intractable.

Students will have the opportunity to gain practical experience with the discussed methods using programming assignments based on Scientific Python.

Keywords

Visual computing, machine learning, numerical linear algebra, numerical analysis, optimization, scientific computing

Learning Prerequisites

Required courses

MATH-101 (Analysis I) and MATH-111 (Linear Algebra).

Recommended courses

Important concepts to start the course

Students are expected to have good familiarity with at least one programming language (e.g. C/C++, Java,

Scala, Python, R, Ruby...). The course itself will rely on Python, but this is straightforward to learn while taking the course.
During the first weeks of the semester, there will be tutorial sessions on using Python and Scientific Python.

Learning Outcomes

By the end of the course, the student must be able to:

- Develop computer programs that use numerical linear algebra and analysis techniques to transform and visualize data.
- Reason about ways of structuring numerical computations efficiently.
- Analyze the numerical stability of programs built on top of floating point arithmetic.
- Recognize numerical problems in visual computing applications and cast them into a form that can be solved or optimized.

Teaching methods

Lectures, interactive demos, theory and programming exercises

Expected student activities

Students are expected to study the provided reading material and actively participate in class and in exercise sessions. They will be given both theoretical exercises and a set of hands-on programming assignments.

Assessment methods

1. Continuous assessment during the semester via project assignments (35%)
2. Final exam (65%)

Resources

Bibliography

Slides and other resource will be provided in class.
The course textbook is
Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics by Justin Solomon (freely available at the following link: http://people.csail.mit.edu/jsolomon/share/book/numerical_book.pdf)
An optional reference is
Scientific Computing: An Introductory Survey (2nd edition) by Michael Heath

Ressources en bibliothèque

- Scientific Computing: An Introductory Survey / Heath
- Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics / Solomon

Moodle Link

- https://go.epfl.ch/CS-328

Prerequisite for

Although it is not a strict prerequisite, this course is highly recommended for students who wish to pursue studies in the area of Visual Computing, in particular: CS-341 (Introduction to computer graphics), CS-440 (Advanced computer graphics), CS-442 (Computer vision), CS-413 (Computational Photography), CS-444 (Virtual Reality), and CS-445 (Digital 3D geometry processing)

CS-302

Parallelism and concurrency in software

Falsafi Babak

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science minor	E	Opt.
Computer science	BA6	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

From sensors,to smart phones,to the world's largest datacenters and supercomputers, parallelism & concurrency is ubiquitous in modern computing.There are also many forms of parallel & concurrent execution in modern platforms with varying degrees of ease of programmability,performance & efficiency.

Content

The goal of this course is to provide a deep understanding of the fundamental principles and trade-offs involved in constructing efficient parallel or concurrent software.

- Performance vs. efficiency
- Forms of parallelism
- Communication models
- Memory models
- Functional parallelism
- Domain-specific languages
- Throughput parallelism
- Data parallelism
- Distributed data parallelism
- Forms of concurrency
- Asynchronous programming
- Coroutines and futures

Learning Prerequisites

Required courses

- CS-200 Computer architecture
- CS-214 Software construction

Recommended courses

CS-202 Computer systems

Important concepts to start the course

• Programming in C/C++, Java or Scala
• Basic assembly language programming
• Basic use of tools to debug

Learning Outcomes

By the end of the course, the student must be able to:

• Construct parallel software
• Construct concurrent software
• Construct efficient software
• Design software for various platforms including CPUs, accelerators and clusters

Teaching methods

• Lectures
• Homework
• Projects

Expected student activities

• Standalone homeworks
• Projects in teams

Assessment methods

• 20% homework
• 30% projects
• 20% midterm
• 30% final

Supervision

Office hours	Yes
Assistants	Yes

Prerequisite for

• CS-471 Advanced multiprocessor architecture
• CS-453 Concurrent computing
• CS-451 Distributed algorithms

PHYS-114

Physique générale : électromagnétisme

Avino Fabio

Cursus	Sem.	Type
HES - IC	H	Opt.
Informatique	BA3	Opt.
Systèmes de communication	BA3	Obl.

Langue d'enseignement	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Le cours traite des concepts de l'électromagnétisme, avec le support d'expériences. Les sujets traités inclus l'électrostatique, le courant électrique et circuits, la magnétostatique, l'induction électromagnétique, les équations de Maxwell, les ondes électromagnétiques, et la lumière comme onde.

Contenu

- Électrostatique : la loi de Coulomb et le champ électrique, la loi de Gauss, le potentiel électrique, capacité et énergie, les champs électriques dans les diélectriques.
- Courant électrique et circuits DC.
- Magnétostatique: les courants comme source du champ d'induction magnétique, les lois fondamentales, les propriétés magnétiques de la matière.
- L'induction électromagnétique : la force électromotrice, la loi d'induction, inductances, l'énergie magnétique, circuits AC.
- Les équations de Maxwell: le courant de déplacement et les équations dans le vide (les ondes électromagnétiques), vecteur de Poynting et énergie EM.
- La lumière comme onde: propgation dans un diélectrique, réflexion, réfraction, ondes stationnaires, interférence et diffraction.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Synthétiser les notions illustrées dans le cours
- Manipuler les équations données dans le cours
- Résoudre des problèmes (pratiques et théoriques) en utilisant les concepts donnés dans le cours
- Choisir ou sélectionner pour résoudre un problème
- Déduire des propriétés à partir de lois fondamentales

Méthode d'enseignement

Ex cathedra avec présentation d'expériences.

Méthode d'évaluation

Examen écrit 100%

Encadrement

Office hours	Oui
Assistants	Oui

Ressources

Bibliographie

D.C. Giancoli, Physics for Scientists and Engineers, Pearson International Edition

Liens Moodle

- https://go.epfl.ch/PHYS-114

COM-302

Principles of digital communications

Telatar Emre

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA6	Obl.
Computer science	BA6	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course is on the foundations of digital communication. The focus is on the transmission problem (rather than being on source coding).

Content

Optimal receiver for vector channels
Optimal receiver for waveform (AWGN) channels
Various signaling schemes and their performance
Efficient signaling via finite-state machines
Efficient decoding via Viterbi algorithm
Communicating over bandlimited AWGN channels
Nyquist Criterion
Communicating over passband AWGN channels

Keywords

Detection, estimation, hypothesis testing, Nyquist, bandwidth, error probability, coding, decoding, baseband, passband, AM, QAM, PSK.

Learning Prerequisites

Required courses

Signal processing for communications and Modèles stochastiques pour les communications

Important concepts to start the course

Linear algebra, probability.

Learning Outcomes

By the end of the course, the student must be able to:

- Estimate the error probability of a communication link
- Design a "physical layer" communication link
- Implement a prototype of a "physical layer" transmitter/receiver via Matlab

Teaching methods

Ex cathedra + exercises + project. Lots of reading at home and exercises in class.

Assessment methods

With continuous control

Resources

Moodle Link

- https://go.epfl.ch/COM-302

Prerequisite for

Advanced Digital Communications
Software-Defined Radio: A Hands-On Course

MATH-232 Probability and statistics (for IC)

Abbé Emmanuel

Cursus	Sem.	Type
Communication systems	BA3	Obl.
Computer science	BA3	Obl.
HES - IC	H	Obl.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

A basic course in probability and statistics

Content

Revision of basic set theory and combinatorics.
Elementary probability: random experiment; probability space; conditional probability; independence.
Random variables: basic notions; density and mass functions; examples including Bernoulli, binomial, geometric, Poisson, uniform, normal; mean, variance, correlation and covariance; moment-generating function; joint distributions, conditional and marginal distributions; transformations.
Many random variables: notions of convergence; laws of large numbers; central limit theorem; delta method; applications.
Statistical inference: different types of estimator and their properties and comparison; confidence intervals; hypothesis testing; likelihood inference and statistical modelling; Bayesian inference and prediction; examples.

Learning Prerequisites

Required courses

Analyse I, II
Algèbre linéaire

Teaching methods

Ex cathedra lectures, exercises and problems

Assessment methods

Written exam

Resources

Notes/Handbook

A polycopié of the course notes, with the problems etc., will be available.

Moodle Link

- https://go.epfl.ch/MATH-232

Prerequisite for

Electrométrie, Théorie du signal, Télécommunications, Information et codage, Fiabilités, ...

EPFL

COM-307 **Projet de recherche en Systèmes de communication I**

Profs divers *

Cursus	Sem.	Type
Systèmes de communication	BA5, BA6	Opt.

Langue d'enseignement	français
Crédits	8
Session	Hiver, Eté
Semestre	Automne
Examen	Pendant le semestre
Charge	240h
Semaines	14
Heures	**2 hebdo**
Projet	2 hebdo
Nombre de places	

Résumé

Travaux de recherche individuelle à effectuer pendant le semestre selon les directives d'une professeure ou d'un professeur ou d'une assistante ou d'un assistant.

Contenu

Sujet de travail à choisir parmi les domaines proposés sur le site web :
Projets par laboratoires

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Organiser un projet
- Evaluer sa progression au cours du projet
- Représenter un projet

Compétences transversales

- Ecrire un rapport scientifique ou technique.
- Ecrire une revue de la littérature qui établit l'état de l'art.

Travail attendu

Rapport écrit à rendre dans le délai imparti.
Les informations relatives au format et au contenu du rapport sont fournies par la superviseuse ou le superviseur du projet.

Méthode d'évaluation

Automne : Le rapport écrit doit être rendu au laboratoire au plus tard l**e vendredi de la seconde semaine** après la fin des enseignements.
Printemps :Le rapport écrit doit être rendu au laboratoire au plus tard **le vendredi de la première semaine** après la fin des enseignements.
La présentation orale est organisée par le laboratoire.

Ressources

Sites web

- https://www.epfl.ch/schools/ic/fr/education-fr/bachelor-fr/projet-semestre/

PHYS-344

Quantum mechanics for non-physicists

Manucharyan Vladimir

Cursus	Sem.	Type
Communication systems	BA5	Obl.
Computer science	BA5	Obl.
Electrical and Electronical Engineering	BA5	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Oral
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course introduces quantum mechanics to students who are interested in pursuing quantum science and technology but have not gone through the standard bachelor physics curriculum. The students will develop quantum intuition by working out numerical examples based on qubits and oscillator systems.

Content

1. Review of classical physics in the context of quantum phenomena
Planetary motion and atoms, radiation and quantization, stochastic processes and interference.
2. Mathematical language of quantum mechanics
Quantum states, operators, matrices, uncertainty, and time-evolution.
3. Basic quantum systems
Particle-in-a-box, harmonic oscillator, anharmonic oscillator, tunneling.
A quick look into stationary perturbation theory.
4. Coupled quantum systems
Entanglement, density matrix, measurement, and decoherence.
A quick look into Fermi's golden rule.
5. Exploring the quantum
Cavity quantum electrodynamics, quantum control, quantum non-demolishing measurements
6. Introduction to quantum computing
(time permitting)

Keywords

Quantum physics
Quantum information
Qubit

Learning Prerequisites

Required courses

Calculus, Linear algebra, Differential equations

Recommended courses

Complex calculus, Mechanics, Electromagnetism

Important concepts to start the course

Complex numbers

Matrices and linear algebra
Familiarity with Python

Learning Outcomes

By the end of the course, the student must be able to:

- Solve basic problems in quantum mechanics
- Manage self-study of modern quantum science

Teaching methods

Lectures and exercises

Expected student activities

Attend lectures and exercise sessions, do the homework

Assessment methods

Oral exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

1) "Feynman's lectures on physics", vol III, selected chapters.
Available online at https://www.feynmanlectures.caltech.edu/III_toc.html
2) "Quantum mechanics: the theoretical minimum" by Lenny Susskind.
Video lectures based on this book are available online
at https://theoreticalminimum.com/courses/quantum-mechanics/2012/winter/lecture-1
3) "An Introduction to Quantum Computing" by Kay, Laflamme, and Mosca (introductory chapters).
Online version available at https://batistalab.com/classes/v572/Mosca.pdf
4) "Exploring the Quantum" by Haroche & Raymond. This book is for advanced students who are interested in learning more material

Moodle Link

- https://go.epfl.ch/PHYS-344

EPFL

CS-290 **Responsible software**

Hardebolle Cécile

Cursus	Sem.	Type
Communication systems	BA3, BA5	Opt.
Computer science	BA3, BA5	Opt.
HES - IC	H	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Software's growing importance increases engineers' responsibility to integrate ethical concerns in the design and development process. This course teaches students concrete strategies for responsible software engineering, focusing on identifying ethical issues and mitigating risks to minimize harms.

Content

The course combines a) knowledge about a range of ethical challenges related to software, and b) pragmatic tools/strategies that students can use in practice to identify and work on ethical risks in the domain of software.
The course will use cases inspired or based on real-world examples to review a range of ethical challenges related to software, including the following themes and questions:

- Safety: How to anticipate potential harmful impacts resulting from the normal and abnormal use of software at the scale of the individuals, groups or society? What types of mechanisms can be used to protect users from harmful impacts and what are their limits?
- Fairness: What is bias in software? What are its sources and what types of impacts can it have? How to identify and mitigate fairness issues?
- Sustainability: What influences the environmental impact of software in terms of energy consumption, CO_2 emission and use of resources? How to estimate and limit this impact?
- Transparency and Autonomy: What issues arise from a lack of transparency in software, and how can methods help users understand software functions and limits? How does software design affect user control and which factors contribute to user empowerment or disempowerment?

Through the course activities, students will get to learn and practice concrete strategies for a responsible approach to software design and development including:

- User and stakeholder analysis strategies
- Strategies for analyzing values
- Impact anticipation strategies
- Risk assessment strategies
- Decision-making strategies

Note: Other concerns, such as security and privacy, are not directly addressed in this course since they are already covered in other courses in the curriculum. However they may be involved in the cases used in the course.

Learning Prerequisites

Required courses

Introduction à la programmation (CS-107)

Important concepts to start the course
Basics of imperative programming

Learning Outcomes

By the end of the course, the student must be able to:
- Assess / Evaluate the level of responsibility of existing software using a set of ethical lenses
- Identify ethical questions during software design and development
- Integrate environmental and social concerns into the software design and development process
- Take into consideration the perspectives of different stakeholders in a software project
- Examine stakeholder values and identify value tensions in a software project
- Investigate the potential benefits and potential harms of software for different stakeholders
- Assess / Evaluate the ethical risks associated with software
- Make design or development decisions taking ethical risks into account

Transversal skills

- Take account of the social and human dimensions of the engineering profession.
- Take responsibility for environmental impacts of her/ his actions and decisions.
- Demonstrate the capacity for critical thinking
- Demonstrate a capacity for creativity.
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

The course is designed with a flipped classroom format, based on online resources from the "Responsible Software" MOOC, developed by the Center for Digital Education.
Each week begins with a problem representative of the topic before introducing the corresponding theoretical content, in the following typical sequence called "problem-solving before instruction":
1. In-class problem-solving on computer (analysis, design or development), with assistants;
2. Independent study of theoretical content (concepts, principles, strategies...) in the form of videos and/or reading material, guided by short activities such as quizzes or open-ended questions;
3. In-class application to one or more concrete case(s) in class with group activities.

Expected student activities

- Complete the exercises
- Study resources (videos and/or readings) and complete preparation activities (quizzes) prior to classroom sessions
- Apply learning to in-class case studies

Assessment methods

- Graded assignments (40%): 2 assignments (20% each) during the semester
- Final exam (60%): 1 individual written test during the semester

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Moodle Link

- https://go.epfl.ch/CS-290

COM-202

Signal processing

Prandoni Paolo, Shkel Yanina

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Communication systems	BA4	Obl.
Computer science	BA4	Opt.
HES - IC	E	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

Signal processing theory and applications: discrete and continuous time signals; Fourier analysis, DFT, DTFT, CTFT, FFT, STFT; linear time invariant systems; filter design and adaptive filtering; sampling; interpolation and quantization; image processing, data communication and control systems.

Content

Signal processing is an engineering discipline that studies how to analyze, modify, and transmit information using mathematical models, practical devices, and numerical algorithms.
The class will cover the following topics:
1. Basic discrete- and continuous-time signals and systems: signal classes and operations on signals, signals as vectors in a vector space
2. Fourier Analysis: properties of Fourier transforms, DFT, DTFT, CTFT; practical Fourier analysis (FFT, STFT)
3. LTI systems: properties and composition, convolution, application of Fourier analysis to LTI System, Laplace and z-Transforms.
4. Analog vs Digital: sampling, interpolation and quantization.
5. Applications: adaptive filtering; image processing, data communication and control systems.

Learning Prerequisites

Required courses

Linear Algebra, Programming (Python), Analysis II

Recommended courses

Analyse III (concurrently), Probability theory (concurrently)

Important concepts to start the course

Vectors and vector space, functions and sequences, infinite series

Learning Outcomes

By the end of the course, the student must be able to:

- Identify signals and signal types
- Describe properties of LTI systems
- Analyze LTI systems by spectral analysis
- Recognize signal processing problems

- Apply the correct analysis tools to specific signals
- Implement signal processing algorithms
- Design digital filters
- Interpret complex signal processing systems

Teaching methods

This course will weave together theoretical analysis in course lectures with practical hands-on labs using Python (via Jupyter notebooks) and more traditional exercise sessions.

Expected student activities

Study class material; complete weekly homework sets (with solutions discussed in subsequent exercise sessions) and participate in Python applied labs.

Assessment methods

The final grade will be almost fully determined by the final exam, with a small grade component based on compilation of weekly laboratory and homework assignments.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/COM-202

CS-214

Software construction

Kuncak Viktor, Odersky Martin, Pit-Claudel Clément

Cursus	Sem.	Type
Communication systems	BA3	Opt.
Computer science minor	H	Opt.
Computer science	BA3	Obl.
Cyber security minor	H	Opt.
Data science minor	H	Opt.
HES - IC	H	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	3 weekly
Number of positions	

Summary

Learn how to design and implement reliable, maintainable, and efficient software using a mix of programming skills (declarative style, higher-order functions, inductive types, parallelism) and fundamental software construction concepts (reusability, abstraction, encapsulation, composition, proofs)

Content

Functional programming:

- Functional programming paradigm
- Recursion and tail-recursion
- Evaluation strategies, lazy evaluation, substitution model
- Modularity, data abstraction, representation independence
- Subtyping, inheritance, type classes
- Polymorphism, variance
- Structural induction
- Stateless parallelism, map-reduce, associative operations
- Effects: state, exceptions
- Documentation, tests, specification
- Interpreters and program semantics

Software engineering:

- Specifications: Documentation, requirements, properties
- Verification: Debugging, tests, monitoring, property-based testing, proofs
- Modularity: Code evolution and refactoring
- Collaboration: Version control, changelogs

Learning Prerequisites

Required courses

Any previous programming course

Recommended courses

CS-107 Introduction à la programmation
CS-108 Pratique de la programmation orientée-objet

Important concepts to start the course

Loops, conditionals, variable and type declarations, computing mathematical expressions

Learning Outcomes

- Implement reliable, efficient, and maintainable software
- Identify data types and operations that lead to computational solutions
- Argue that an solution is correct
- Transform a program to change its behavior in a desirable way
- Design and implement data-parallel software using parallel collections

Teaching methods

- Ex cathedra (live lectures)
- Recorded videos
- Exercise and lab sessions
- Online discussions

Expected student activities

- Attending lectures
- Watching and understanding recorded videos
- Solving exercises individually or in groups
- Completing individual graded programming assignments (labs / mini-projects)
- Completing midterm and end-of-semester exams

Assessment methods

- **30%** Midterm exam during the semester
- **40%** Final exam during the exam session
- **30%** Programming assignments (labs)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Moodle Link

- https://go.epfl.ch/CS-214

Prerequisite for

CS-320 Computer language processing
CS-311 The Software enterprise - from ideas to products
CS-452 Foundations of software
CS-550 Formal verification

EE-310

Systèmes embarqués microprogrammés

Atienza Alonso David

Cursus	Sem.	Type
Génie mécanique	MA1, MA3	Opt.
Génie électrique et électronique	BA5	Obl.
HES - EL	H	Opt.
Informatique	BA5	Opt.
Mineur en Technologies spatiales	H	Opt.
Systèmes de communication	BA5	Opt.

Langue d'enseignement	français
Crédits	4
Retrait	Non autorisé
Session	Hiver
Semestre	Automne
Examen	Oral
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Projet	2 hebdo
Nombre de places	**80**

Il n'est pas autorisé de se retirer de cette matière après le délai d'inscription.

Résumé

L'étudiant comprendra les architectures des systèmes embarqués microprogrammés, les architectures des microprocesseurs, hiérarchie de mémoire et les différents périphériques de Entrée/Sortie (E/S) inclus, utilisant comme étude de cas la plate-forme portable Nintendo DS.

Contenu

Introduction aux systèmes embarqués microprogrammés
- Introduction à la plate-forme physique de la Nintendo DS
- Introduction au logiciel de compilation et GUI

Microprocesseur et hiérarchie de mémoire
- Architecture des microprocesseurs dans la Nintendo DS
- Organisation de la hiérarchie de mémoire et bus
- Programmation combinée C-assembleur
- DMA et gestion de la mémoire

Gestion de l'E/S et des circuits périphériques
- Description des périphériques et de l'E/S dans la Nintendo DS
- Contrôleurs d'interruption simple
- Interface et gestion des boutons
- Interface et gestion du clavier
- Interface pour dessiner sur l'écran LCD et sa gestion
- Interface et gestion de l'écran tactile
- Interface et gestion du son

Développement des jeux avec la Nintendo DS
- Division des fonctions des jeux
- Contrôleurs d'interruption complexe
- Programmation avancée des graphiques
- Connexion sans fils (Bluetooth, WiFi)

Mots-clés

systèmes embarqués, microprocesseurs, multi-core, gestion de E/S, développement des systèmes, contrôleurs d'interruption complexe

Compétences requises

Cours prérequis indicatifs
Projet de programmation, Systèmes microprogrammés

Concepts importants à maîtriser
Programmation, architectures des microcontrolleurs

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Discriminer entre les types de dispositifs d'E/S.
- Créer un projet de l'architecture complet d'un système embarqué avec la plateforme Nintendo DS.
- Décrire les conditions de l'interaction de différents types de périphériques d'E/S.
- Implémenter des logiciels pour contrôler les périphériques d' E/S.
- Développer des projets collaboratifs au sein d'une équipe d'étudiants.

Compétences transversales

- Utiliser une méthodologie de travail appropriée, organiser un/son travail.
- Auto-évaluer son niveau de compétence acquise et planifier ses prochains objectifs d'apprentissage.
- Fixer des objectifs et concevoir un plan d'action pour les atteindre.
- Gérer ses priorités.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.

Méthode d'enseignement

Ex cathedra, et TP individuel et par groupes

Travail attendu

Participation au cours, résolution des exercices de manière individuelle et réalisation d'un projet en groupe de deux personnes.

Méthode d'évaluation

Evaluations pratiques, midterm exam (35% de la note finale) durant le semestre, examen oral à la fin du semestre (65% de la note finale)

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie
Notes de cours - Polycopié,
Support et liste de références distribués en cours, cf. URL du cours

Polycopiés
Polycopié disponible au Rolex Learning Center - "Course Notes"

Liens Moodle

- https://go.epfl.ch/EE-310

Préparation pour

- Systèmes numériques (EE-334)
- TP de conception de systèmes numériques (EE-397)

CS-234 **Technologies for democratic society**

Ford Bryan Alexander

Cursus	Sem.	Type
Communication systems	BA3	Opt.
Computer science	BA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This course will offer students a broad but hands-on introduction to technologies of human self-organization.

Content

The course will present students with a view of self-organization technologies set in a long-term historical perspective, extending from their roots in ancients principles of democracy and governance, up to recent high-tech innovation such as social networking, e-voting, blockchains, and delegative democracy. The course will cover the many fundamental organization challenges these technologies attempt to address, such as :

- Coordination : do participants communicate in person, electronically, or by passing secret notes ?
- Membership : who has the right to participate as a member or citizen ? Can membership be faked ?
- Equity or fairness : how much power or weight does each participant have ? Can weight be hacked ?
- Filtering : how to separate signal from noise, real expertise from appealing bluster ?
- Scalability : does the self-organizing technology work for only 10 members, or 100? 1000 ? 1 M ? 1 B ?
- Integrity : how does self-organizing technology prevent hacking or tampering by malicious parties ?
- Self-determination : does the technology protect freedoms such as expression and association ?
- Privacy : what acts of participation does the technology keep private, and what are considered public?
- Representation : is participation direct or representative ? How are representatives chosen ?
- Accountability : how are participants and/or representatives kept accountable for their actions ?
- Transparency : does the technology allow participants to verify that it is operating correctly ? How ?
- Incentives : how does the technology encourage or incentivize people to use it, for good or ill ?
- Psychology : how does the technology interplay with the unique properties of the human mind ?

Learning Prerequisites

Important concepts to start the course

Basic computing and programming skills

Learning Outcomes

By the end of the course, the student must be able to:

- Explore technologies available for societal self-organization
- Expound key challenges and risks in using these technologies

- Discuss social implications of digital communication and organization technologies

Teaching methods

The course will use readings, discussions, and exercises to lead students through an exploration of the vast number of different technological approaches to these challenges and issues, from extremely low-tech (e.g., picking representatives by drawing straws) to the latest experimental technologies. In different weeks the students will explore hands-on the architecture, design, practical use, and strenghts and weaknesses of different self-organization technologies, such as :

- Public discussion forums such as UseNet, Twitter, and Reddit
- Community self-organization systems such as Loomio
- Peer review systems such as HotCRP
- E-voting systems in use in around the world (especially the US and Switzerland)
- Experimental participatory delegative democracy systems such as LiquidFeedback
- Cryptocurrencies and smart contract systems such as Bitcoin and Ethereum

The course work will involve a substantial amount of reading background materials, both technical and non-technical and from a variety of disciplines including computer science, social science, political science, and law. The lectures will be heavily discussion-oriented, covering both the background readings and hands-on exercises in addition to material presented in the lectures.

Expected student activities

The course will encourage students to "learn by doing" through exercises with practical systems. Students will be required to use some of these systems in groups in "hands-on" self-organization exercises, to get firsthand comparative experience of how they work, and in what ways they succeed and fail.

Assessment methods

Students will be assessed through regular exercises and mini-quizzes, participation in "peer review" activities, a small project in the second half of the semester on which the students must report, and a written final exam. Grading will be based substantially on demonstrated active participation in the deliberative course exercises, in addition to learning and understanding of the course content itself.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-234

CS-311

The software enterprise - from ideas to products

Bugnion Edouard, Candea George

Cursus	Sem.	Type
Communication systems	BA5	Opt.
Computer science minor	H	Opt.
Computer science	BA5	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**13 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	10 weekly
Number of positions	

Remark

Special schedule

Summary

This course teaches the journey taken by software engineering teams from incipient ideas to software products that solve real problems for real people.

Content

The combination of technical and product-management skills acquired in this course will enable students to build effective software products in teams, either within an existing organization or as founders of their own startups.

- Requirements and specifications
- Validation, testing, and debugging
- DevOps (version control, project management, issue tracking, continuous integration)
- Behavior-driven and test-driven development
- Development processes
- Cloud-platform and mobile-platform architectures
- Product architecture
- Security, privacy, and data protection
- Scaling to millions of users
- Differentiation and value proposition/opportunity assessment
- MVP and product roadmap
- Business model alternatives
- Intellectual property and open-source software/hardware

Learning Prerequisites

Required courses

CS-173 Fundamentals of Digital Systems (BA2)
CS-214 Software construction (BA3)
CS-202 Computer systems (BA4)

Important concepts to start the course

Must be proficient in programming in C and Python and object-oriented Java/Kotlin/Scala

Learning Outcomes

By the end of the course, the student must be able to:

- Design and implement mobile and/or cloud apps
- Master a variety of system design patterns
- Work in and manage a team of developers
- Identify opportunities for using software to solve real-world problems
- Plan a software product from A-to-Z
- Assess / Evaluate progress against the plan, and adapt the plan as appropriate
- Manage priorities & basics of product management
- Optimize the use of time and resources to achieve a given goal
- Take feedback (critique) and respond in an appropriate manner
- Develop auto-didact skills

Teaching methods

- Ex cathedra
- Recitations and workshops
- Extensive team-based project

Expected student activities

- Work with team members to complete a substantial project
- Independently research solutions, study documentation, etc. (auto-didact)

Assessment methods

- Throughout the semester (continuous control)
- Grade determined based on both team and individual performance in the project
- Deliverables include an implemented software product v.1 and a written product plan for v.2

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-311

EPFL

CS-251

Theory of computation

Göös Mika

Cursus	Sem.	Type
Communication systems	BA6	Opt.
Computer science minor	E	Opt.
Computer science	BA6	Obl.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course constitutes an introduction to theory of computation. It discusses the basic theoretical models of computing (finite automata, Turing machine), as well as, provides a solid and mathematically precise understanding of their fundamental capabilities and limitations.

Content

- Basic models of computation (finite automata, Turing machine)
- Elements of computability theory (undecidability, reducibility)
- Introduction to time complexity theory (P, NP and theory of NP-completeness)

Keywords

theory of computation, Turing machines, P vs. NP problem, complexity theory, computability theory, finite automata, NP-completeness

Learning Prerequisites

Required courses

CS-101 Advanced information, computation, communication I
CS-250 Algorithms

Learning Outcomes

By the end of the course, the student must be able to:

- Perform a rigorous study of performance of an algorithm or a protocol
- Classify computational difficulty of a decision problem
- Define the notion of NP-completeness
- Analyze various computation models
- Design a reduction between two computational problems
- Characterize different complexity classes
- Explain P vs. NP problem

Transversal skills

- Use a work methodology appropriate to the task.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Ex cathedra with exercises

Assessment methods

Written exam and continuous control

Resources

Moodle Link

- https://go.epfl.ch/CS-251

CS-500

AI product management

Kaboli Amin, Zamir Amir

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**7 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	3 weekly
Number of positions	

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The course focuses on the development of real-word AI/ML products. It is intended for students who have acquired a theoretical background in AI/ML and are interested in applying that toward developing AI/ML-oriented products.

Content

AI is set to transform several industry sectors, and there is high demand for AI product managers. AI product management is a complex role that requires an understanding of both AI and product management. This course will enable students to identify opportunities for developing new AI products, understand when they should use AI in an existing product/process, manage the development of AI products, and launch AI products successfully. The lectures will introduce general product management to the students, and the guest lectures, by leading figures in AI industries, explain how the general product management skills are applied to the development of AI products.

Module 1: Introduction to AI product management

- Why is this needed?
- Product strategy
- Setting product objectives & identifying opportunities

Module 2: AI Product Discovery

- Understanding customers and problems
- Creating and testing hypothesis
- Defining product requirements
- Establishing product-market fit
- Setting AI product vision, strategy, roadmap

Module 3: AI Product development

- Mastering agile methodologies
- AI product design, tests, and development
- Managing team dynamics and effective communication with stakeholders

Module 4: AI Product Delivery

- Planning and executing a successful AI product launch
- Marketing the AI product
- Release and performance metrics for continuous improvement of AI products

Keywords

Artificial Intelligence (AI), AI product managers, Innovation

Learning Prerequisites

Required courses

CS-233 Introduction to machine learning or CS-433 Machinie learning or equivalent course on the basics of machine learning and deep learning

Important concepts to start the course

- Python programming
- Bascis of deep learning and machine learning
- Basics of probability and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- and understand opportunities for an AI product or using AI within an existing product
- the development of AI features
- Launch AI products successfully

Transversal skills

- Demonstrate the capacity for critical thinking
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Communicate effectively, being understood, including across different languages and cultures.
- Set objectives and design an action plan to reach those objectives.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Resolve conflicts in ways that are productive for the task and the people concerned.
- Make an oral presentation.
- Take account of the social and human dimensions of the engineering profession.

Teaching methods

- Formal lectures
- Group activities
- Class discussions
- Simulation games
- Hands-on exercises
- Project-based learning
- Real-world case studies

- Guest lectures by leading academic and industry figuers

Expected student activities

- **Individual** : Case evaluations, self-study, class discussions
- **In-group** : In-class exercises, projects, simulations games
- **Presentation** : Weekly presentations of assignments in coaching sessions

Assessment methods

Continuous evaluation of case reports, proejcts, individual and group presentations, class discussions, during the semester. More precisely :
25% Weekly in-class work and engagement
45% Class assignments, presentations, projects, and case reports
30% Final (final report and presentation and understanding of the case)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Cagan, M. (2017). *How to Create Tech Products Customers Love.* Wiley
- Kahneman, D., Sibony, O., & Sunstein, C. R. (2021). *Noise: A flaw in human judgment*. Little, Brown.
- Iansiti, M., & Lakhani, K. R. (2020). *Competing in the age of AI: strategy and leadership when algorithms and networks run the world*. Harvard Business Press.

Ressources en bibliothèque

- How to Create Tech Products Customers Love / Cagan
- Noise / Kahneman
- Competing in the age of AI / Iansiti

Moodle Link

- https://go.epfl.ch/CS-500

Advanced compiler construction

CS-420

Schinz Michel

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Students learn several implementation techniques for modern functional and object-oriented programming languages. They put some of them into practice by developing key parts of a compiler and run time system for a simple functional programming language.

Content

Part 1: implementation of high-level concepts

- functional languages: closures, continuations, tail call elimination,
- object-oriented languages: object layout, method dispatch, membership test.

Part 2: optimizations

- compiler intermediate representations (RTL, SSA, CPS),
- inlining and simple optimizations,
- register allocation.

Part 3: run time support

- interpreters and virtual machines,
- memory management (including garbage collection).

Keywords

compilation, programming languages, functional programming languages, object-oriented programming languages, code optimization, register allocation, garbage collection, virtual machines, interpreters, Scala.

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Excellent knowledge of Scala and C programming languages

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the quality of a compiler intermediate representation
- Design compilers and run time systems for object-oriented and functional programming languages
- Implement rewriting-based compiler optimizations
- Implement efficient virtual machines and interpreters
- Implement mark and sweep or copying garbage collectors

Teaching methods

Ex Cathedra, mini-project

Assessment methods

Continuous control (mini-project 80%, final exam 20%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

The Garbage Collection Handbook - The Art of Automatic Memory Management, second edition, Richard Jones, Antony Hosking, Eliot Moss (ISBN 9781032218038).

Ressources en bibliothèque

- The Garbage Collection Handbook / Jones

Websites

- https://cs420.epfl.ch/

CS-470

Advanced computer architecture

Ienne Paolo

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Labs	2 weekly
Number of positions	

Summary

The course studies techniques to exploit Instruction-Level Parallelism (ILP) statically and dynamically. It also addresses some aspects of the design of domain-specific accelerators. Finally, it explores security challenges based on microarchitectural features and hardware isolation techniques.

Content

Pushing processor performance to its limits:

- Principles of Instruction Level Parallelism (ILP)
- Register renaming techniques
- Prediction and speculation
- Simultaneous multithreading
- VLIW and compiler techniques for ILP
- Dynamic binary translation

Domain specific architectures and accelerators:

- Specificities of embedded vs. general computing processors
- Overview of DSPs and related compilation challenges
- High-Level Synthesis and accelerators

Hardware security:

- Information leakage through the microarchitecture
- Trusted Execution Environments
- Physical side-channel attacks

Keywords

Processors, Instruction Level Parallelism, Systems-on-Chip, Embedded Systems, High-Level Synthesis, Hardware Security.

Learning Prerequisites

Required courses

• CS-208 Computer Architecture I

Recommended courses

• CS-209 Computer Architecture II

Important concepts to start the course

Undergraduate knowledge of digital circuit design and of computer architecture

Learning Outcomes

By the end of the course, the student must be able to:

- Design strategies to exploit instruction level parallelism in processors.
- Contrast static and dynamic techniques for instruction level parallelism.
- Design effective processor (micro-)architectures for which efficient compilers can be written.
- Develop hardware accelerators competitive to best commercial processors
- Defend against security threats based on microarchitectural processor features

Teaching methods

Courses, labs, and compulsory homeworks.

Assessment methods

Homeworks (30%)
Final exam (70%)

Supervision

Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

• John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufman, 6th edition, 2017.

Ressources en bibliothèque

• Computer Architecture / Hennessy

Moodle Link

• https://go.epfl.ch/CS-470

Prerequisite for

• CS-471 Advanced Multiprocessor Architecture

CS-440

Advanced computer graphics

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	1 weekly
Project	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

This course covers advanced 3D graphics techniques for realistic image synthesis. Students will learn how light interacts with objects in our world, and how to recreate these phenomena in a computer simulation to create synthetic images that are indistinguishable from photographs.

Content

This is a project-based course: students will initially receive a basic software package that lacks most rendering-related functionality.
Over the course of the semester, we will discuss a variety of concepts and tools including the basic physical quantities, how light interacts with surfaces, and how to solve the resulting mathematical problem numerically to create realistic images. Advanced topics include participating media, material models for sub-surface light transport, and Markov Chain Monte Carlo Methods.
Each major topic is accompanied by an assignment so that students can implement solution algorithms and obtain practical experience with these techniques within their own software framework.
Towards the end of the course, students will realize a self-directed final project that extends their rendering software with additional features of their own choosing. The objective of the final project is to create a single image of both technical and artistic merit that is entered into a rendering competition and judged by an independent panel of computer graphics experts.

Learning Prerequisites

Required courses

Nothing

Recommended courses

Introduction to Computer Graphics

Important concepts to start the course

We will rely on calculus, linear algebra and use basic concepts of algorithms and data structures. Students are expected to be familiar with the C++ programming language that is used in the programming assignments.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize and understand the physical quantities of light transport and be able to perform basic computations using pencil+paper
- Explain a range of surface and subsurface material models
- Explain the rendering and radiative transfer equation and show how to construct Monte Carlo methods to solve them
- Design and implement an advanced rendering system based on Monte Carlo integration
- Assess / Evaluate the performance and conceptual limits of the implemented simulation code

Teaching methods

Lectures, interactive demos, theory and programming exercises, programming project, project tutoring

Expected student activities

The student are expected to study the provided reading material and actively participate in class. They should prepare and resolve the exercises, prepare and carry out the programming project.

Assessment methods

Intermediate assignments (40%), final project (60%)

Resources

Bibliography

A list of books will be provided at the beginning of the class

Notes/Handbook

Slides and online resources will be provided in class

Websites

- https://rgl.epfl.ch/courses/ACG22

Moodle Link

- https://go.epfl.ch/CS-440

COM-501

Advanced cryptography

Vaudenay Serge

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course reviews some failure cases in public-key cryptography. It introduces some cryptanalysis techniques. It also presents fundamentals in cryptography such as interactive proofs. Finally, it presents some techniques to validate the security of cryptographic primitives.

Content

1. **The cryptograhic zoo**: definitions, cryptographic primitives, math, algorithms, complexity
2. **Cryptographic security models:** security notions for encryption and authentication, game reduction techniques, RSA and Diffie-Hellman security notions
3. **Public-key cryptanalysis:** side channels, low RSA exponents, discrete logarithm, ElGamal signature
4. **Interactive proofs:** NP-completeness, interactive systems, zero-knowledge
5. **Symmetric-key cryptanalysis:** differential and linear cryptanalysis, hypothesis testing, decorrelation
6. **Proof techniques:** random oracles, leftover-hash lemma, Fujisaki-Okamoto transform

Keywords

cryptography, cryptanalysis, interactive proof, security proof

Learning Prerequisites

Required courses

- Cryptography and security (COM-401)

Important concepts to start the course

- Cryptography
- Mathematical reasoning
- Number theory and probability theory
- Algorithmics
- Complexity

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the security deployed by cryptographic schemes

- Prove or disprove security
- Justify the elements of cryptographic schemes
- Analyze cryptographic schemes
- Implement attack methods
- Model security notions

Teaching methods

ex-cathedra

Expected student activities

- active participation during the course
- take notes during the course
- do the exercises during the exercise sessions
- complete the regular tests and homework
- read the material from the course
- self-train using the provided material
- do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

- homework (30%)
- regular graded tests (30%)
- midterm exam (40%)

Final exam averaged (same weight) with the contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

- Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
- A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.
- Algorithmic cryptanalysis. Antoine Joux. CRC 2009.

Ressources en bibliothèque

- Algorithmic cryptanalysis / Joux
- Communication security / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-501

Videos

- https://mediaspace.epfl.ch/channel/COM-501+Advanced+Cryptography

CS-471 Advanced multiprocessor architecture

Falsafi Babak

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**12 weekly**
Lecture	4 weekly
Project	8 weekly
Number of positions	

Summary

Multiprocessors are basic building blocks for all computer systems. This course covers the architecture and organization of modern multiprocessors, prevalent accelerators (e.g., GPU, TPU), and datacenters. It includes a research project on multiprocessors and post-Moore era datacenters.

Content

- Introduction
- Metrics and methodologies
- Parallel programming models
- Communication models
- Applications and Workloads
- Cache hierarchies & memory models
- Memory & storage hierarchies
- Interconnects
- GPUs
- AI/ML/Analytic accelerators
- Near-memory computing
- Datacenters & cloud Computing
- Cloud-native CPU
- Cloud-native memory hierarchies
- Sustainable architecture

Learning Prerequisites

Recommended courses

- Advanced computer architecture
- Systems for data management and data science

Learning Outcomes

- Explore the development trend of computation systems and datacenters
- Establish the basic model to analyze the performance and characteristics of foundational workloads operating in cloud environments.
- Classify and describe the components of modern parallel systems, including multiple processors, cache hierarchies, memory systems, interconnects, and accelerators, and their roles in handling emerging workloads
- Define and clarify research questions and opportunities
- Interpret and critique research papers and extract insights for research questions
- Plan and conduct a research project
- Present research contributions

Teaching methods

Lecture, research paper retrieval, and a research project

Assessment methods

- Homework: 10%
- Research project (in group): 40%
- Midterm exam: 20%
- Final exam: 30%

Resources

Websites

- https://parsa.epfl.ch/course-info/cs471/

Advanced operating systems

CS-477

Kashyap Sanidhya

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

This course teaches advanced OS system design. Using an extensive hands-on approach, the course focuses on traditional and new concepts proposed in the past two decades in the area of operating systems.

Content

- Concurrent execution
- Memory management using things like virtual memory and memory allocations
- Scalability, such as lock-free data structures
- File systems
- Operating system architecture
- Virtualization
- Security such as data security, integrity, and authentification

Learning Prerequisites

Required courses

CS-200 Computer architecture
CS-202 Computer systems (or COM-208 Computer networks / CS-323 Introduction to operating systems)
CS-214 Software construction (or CS-210 Functional programming)
CS-300 Data-Intensive Systems (or CS-322 Introduction to database systems)

Important concepts to start the course

This course is a hands-on course that required solid background in operating systems, databases, programming language, and computer architecture.

Learning Outcomes

By the end of the course, the student must be able to:

- Identify and manage key components of operating systems
- Choose or critique design choices for OS
- Create and design an OS with practical choices
- Report performance and possible optimizations for applications

Teaching methods

- In-class lectures
- Homework assignments
- Weekly preparatory exercises
- Programming exercise

Expected student activities

- Attend lectures and tutorial sessions
- Complete the homework assignments
- Participate actively in the course
- Complete programming exercises

Assessment methods

- Preparatory exercise 10%
- Labs 40%
- Final exam 50%

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-477

COM-417

Advanced probability and applications

Shkel Yanina

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Obl.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Electrical Engineering		Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course, various aspects of probability theory are considered. The first part is devoted to the main theorems in the field (law of large numbers, central limit theorem, concentration inequalities), while the second part focuses on the theory of martingales in discrete time.

Content

- sigma-fields, random variables
- probability measures, distributions
- independence, convolution
- expectation, characteristic function
- random vectors and Gaussian random vectors
- inequalities, convergences of sequences of random variables
- laws of large numbers, applications and extensions
- convergence in distribution, central limit theorem and applications
- moments and Carleman's theorem
- concentration inequalities
- conditional expectation
- martingales, stopping times
- martingale convergence theorems

Keywords

probability theory, measure theory, martingales, convergence theorems

Learning Prerequisites

Required courses

Basic probability course
Calculus courses

Recommended courses

Complex analysis

Important concepts to start the course

This course is NOT an introductory course on probability: the students should have a good understanding and practice of basic probability concepts such as: distribution, expectation, variance, independence, conditional probability.
The students should also be at ease with calculus. Complex analyisis is a plus, but is not required.

On the other hand, no prior background on measure theory is needed for this course: we will go through the basic concepts one by one at the beginning.

Learning Outcomes

By the end of the course, the student must be able to:

- understand the main ideas at the heart of probability theory

Teaching methods

Ex cathedra and flipped lectures + exercise sessions

Expected student activities

active participation to exercise sessions

Assessment methods

graded homeworks 20%
midterm 20%
final exam 60%

Resources

Bibliography

Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability,1st edition, www.ProbabilityBookstore.com, 2007.
Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory,2nd edition, World Scientific, 2006.
Geoffrey R. Grimmett, David R. Stirzaker, Probability and Random Processes,3rd edition, Oxford University Press, 2001.
Richard Durrett, Probability: Theory and Examples, 4th edition, Cambridge University Press, 2010.
Patrick Billingsley, Probability and Measure, 3rd edition, Wiley, 1995.

Ressources en bibliothèque

- Probability and Random Processes
- Sheldon M. Ross, Erol A. Pekoz, A Second Course in Probability, 1st ed
- Patrick Billingsley, Probability and Measure, 3rd ed
- Richard Durrett, Probability: Theory and Examples, 4th ed
- Jeffrey S. Rosenthal, A First Look at Rigorous Probability Theory, 2nd ed

Notes/Handbook

available on the course website

Websites

- https://moodle.epfl.ch/course/view.php?id=14557

Moodle Link

- https://go.epfl.ch/COM-417

Prerequisite for

Advanced classes requiring a good knowledge of probability

EPFL

CS-523

Advanced topics on privacy enhancing technologies

Troncoso Carmela

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This advanced course will provide students with the knowledge to tackle the design of privacy-preserving ICT systems. Students will learn about existing technologies to prect privacy, and how to evaluate the protection they provide.

Content

The course will cover the following topics :
- Privacy definitions and concepts
- Privacy-preserving cryptographics solutions : anonymous credentials, zero-knowledge proofs, secure multi-party computation, homomorphic encryption, Private information retrieval (PIR), Oblivious RAM (ORAM)
- Anonymization and data hiding : generalization, differential privacy, etc
- Machine learning and privacy
- Protection of metadata : anonymous communications systems, location privacy, censorpship resistance
- Online tracking and countermeasures
- Privacy engineering : design and evaluation (evaluation metrics and notions)
- Legal aspects of privacy

Keywords

Privacy, anonymity, homomorphic encryption, secure multi-paty computation, anonymous credentials, ethics

Learning Prerequisites

Required courses

COM-301 Computer security
COM-402 Information security and privacy

Recommended courses

COM-401 Cryptography and security

Important concepts to start the course

Basic programming skills; basics of probabilities and statistics; basics of cryptography

Learning Outcomes

By the end of the course, the student must be able to:

- Select appropriately appropriately privacy mechanisms

- Develop privacy technologies
- Assess / Evaluate privacy protection
- Reason about privacy concerns

Teaching methods

Lectures and written exercises to deepen understanding of concepts
Programming-oriented assignments to practice use of privacy technologies

Expected student activities

Participation in the lectures. Active participation is encouraged.
Participation in exercise session and complete the exercises regularly
Completion of programming assignments

Supervision

Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-523

CS-450

Algorithms II

Kapralov Michael, Svensson Ola Nils Anders

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**7 weekly**
Lecture	4 weekly
Exercises	3 weekly
Number of positions	

Summary

A first graduate course in algorithms, this course assumes minimal background, but moves rapidly. The objective is to learn the main techniques of algorithm analysis and design, while building a repertory of basic algorithmic solutions to problems in many domains.

Content

Algorithm analysis techniques: worst-case and amortized, average-case, randomized, competitive, approximation. Basic algorithm design techniques: greedy, iterative, incremental, divide-and-conquer, dynamic programming, randomization, linear programming. Examples from graph theory, linear algebra, geometry, operations research, and finance.

Learning Prerequisites

Required courses

An undergraduate course in Discrete Structures / Discrete Mathematics, covering formal notation (sets, propositional logic, quantifiers), proof methods (derivation, contradiction, induction), enumeration of choices and other basic combinatorial techniques, graphs and simple results on graphs (cycles, paths, spanning trees, cliques, coloring, etc.).

Recommended courses

An undergraduate course in Data Structures and Algorithms.
An undergraduate course in Probability and Statistics.

Important concepts to start the course

Basic data structures (arrays, lists, stacks, queues,trees) and algorithms (binary search; sorting; graph connectivity); basic discrete mathematics (proof methods, induction, enumeration and counting, graphs); elementary probability and statistics (random variables, distributions, independence, conditional probabilities); data abstraction.

Learning Outcomes

By the end of the course, the student must be able to:

- Use a suitable analysis method for any given algorithm
- Prove correctness and running-time bounds
- Design new algorithms for variations of problems studied in class
- Select appropriately an algorithmic paradigm for the problem at hand

- Define formally an algorithmic problem

Teaching methods

Ex cathedra lecture, reading

Assessment methods

- midterm (30%)
- homework (30%)
- final exam (40%)

Supervision

Forum	Yes
Others	For details, see the course web page

Resources

Websites

- http://theory.epfl.ch/courses/AdvAlg/

Moodle Link

- https://go.epfl.ch/CS-450

EE-512

Applied biomedical signal processing

Lemay Mathieu

Cursus	Sem.	Type
Biomedical technologies minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The goal of this course is twofold: (1) to introduce physiological basis, signal acquisition solutions (sensors) and state-of-the-art signal processing techniques, and (2) to propose concrete examples of applications for vital sign monitoring and diagnosis purposes.

Content

- Introduction on the basics in anatomy and physiology of autonomous nervous system, electrical cardiac system, hemodynamic basis, brain and respiratory activities as well as location.
- Digital signal processing basics including sampling, Fourier transform, filtering, stochastic signal correlation and power spectral density. Time-frequency analysis including short-term Fourier and wavelet transforms. Linear modelling including autoregressive models, linear prediction, parametric spectral estimation, and criteria for model selection. Adaptive filtering including adaptive prediction and estimations of transfer functions as well as adaptive interference cancellation.
- Digital signal processing miscellaneous techniques including polynomial models, singular value decomposition and principal component analysis, phase-rectified signal averaging, source separation, support vector regression, and neural network structures such as CNN and RNN.
- Applications and exercises related to cardiac arrythmia detection and classification, central blood pressure estimation, sleep phase classification, heart rate tracking robust against motion artefacts, epilepsy event detection, fall detection, apnoea detection, SpO2 estimation, and respiration tracking and volume estimation. These exercises will be based on biomedical signals such as bio-impedance, electrocardiogram, electroencephalogram, hypnogram, movement (accelerometer, gyroscope, and barometer), photoplethysmography, vocal/audio.

Keywords

signal processing, biomedical engineering, signal modelling, spectral analysis, adaptive filtering, algorithm design

Learning Prerequisites

Recommended courses

Signal processing for telecommunications COM-303
Signal processing EE-350

Important concepts to start the course

basics of discrete-time signal analysis
basics in signal processing programming

Teaching methods

Ex cathedra lectures (approx.. 2h per module) and practical work using Matlab/Python (approx.. 2h per module). The student should provide a separate report for each of the practical work session for evaluation. Grades are based on the practicals and a final exam.

Expected student activities

- Attending lectures
- Processing and analysing human data
- Testing signal processing techniques

Assessment methods

1.75 points in total for the lab/exercise sessions reports during the semester (35% of the final total grade)
3.25 points for the final exam during the examination period (65% of the final total grade)

Supervision

Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/EE-512

MATH-493

Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Ing.-math	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course covers topics in applied biostatistics, with an emphasis on practical aspects of data analysis using R statistical software. Topics include types of studies and their design and analysis, high dimensional data analysis (genetic/genomic) and other topics as time and interest permit.

Content

Types of studies
Design and analysis of studies
R statistical software
Reproducible research techniques and tools
Report writing
Exploratory data analysis
Liniear modeling (regression, anova)
Generalized linear modeling (logistic, Poission)
Survival analysis
Discrete data analysis
Meta-analysis
High dimensional data analysis (genetics/genomics applications)
Additional topics as time and interest permit

Keywords

Data analysis, reproducible research, statistical methods, R, biostatistical data analysis, statistical data analysis

Learning Prerequisites

Required courses

This course will be very difficult for students with no previous course or experience with statistics. Previous experience with R is neither assumed nor required.

Recommended courses

Undergraduate statistics course

Important concepts to start the course

It is useful to review statistical hypothesis testing.

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret analysis results
- Justify analysis plan
- Plan analysis for a given dataset
- Analyze various types of biostatistical data
- Synthesize analysis into a written report
- Report plan of analysis and results obtained
- Synthesize analysis into a written report
- Report plan of analysis and results obtained
- Justify analysis plan
- Plan analysis for a given dataset
- Interpret analysis results
- Analyze various types of biostatistical data

Transversal skills

- Write a scientific or technical report.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Take feedback (critique) and respond in an appropriate manner.
- Use a work methodology appropriate to the task.

Teaching methods

Lectures and practical exercises using R. Typically, each week covers an analysis method in the lecture and then the corresponding exercise session consists of an R practical showing how to implement the methods using R. In each practical, students use R to carry out analyses of the relevant data type for that week.

Expected student activities

Students are expected to participate in their learning by attending lectures and practical exercise sessions, posing questions, proposing topics of interest, peer reviewing of preliminary reports, and interacting with teaching staff regarding their understanding of course material. In addition, there will be a number of short activities in class aimed at improving English for report writing.

Assessment methods

Evaluation is based on written reports of projects analyzing biostatistical data.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

To be provided during the course.
Pre-recorded lectures (videos) will also be provided.

Moodle Link

- https://go.epfl.ch/MATH-493

EPFL

CS-401

Applied data analysis

Brbic Maria

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data and Internet of Things minor	H	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Obl.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Statistics	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course teaches the basic techniques, methodologies, and practical skills required to draw meaningful insights from a variety of data, with the help of the most acclaimed software tools in the data science world (pandas, scikit-learn, Spark, etc.)

Content

Thanks to modern software tools that allow to easily process and analyze data at scale, we are now able to extract invaluable insights from the vast amount of data generated daily. As a result, both the business and scientific world are undergoing a revolution which is fueled by one of the most sought after job profiles: the data scientist.
This course covers the fundamental steps of the data science pipeline:
Data wrangling

- Data acqusition (scraping, crawling, parsing, etc.)
- Data manipulation, array programming, dataframes

- The many sources of data problems (and how to fix them): missing data, incorrect data, inconsistent representations
- Data quality testing with crowdsourcing

Data interpretation

- Statistics in practice (distribution fitting, statistical significance, etc.)
- Working with "found data" (design of observational studies, regression analysis)
- Machine learning in practice (supervised and unsupervised, feature engineering, evaluation, etc.)
- Text mining: preprocessing steps, vector space model, topic models
- Social network analysis (properties of real networks, working graph data, etc.)

Data visualization

- Introduction to different plot types (1, 2, and 3 variables), layout best practices, network and geographical data
- Visualization to diagnose data problems, scaling visualization to large datasets, visualizing uncertain data

Reporting

- Results reporting, infographics
- How to publish reproducible results

The students will learn the techniques during the ex-cathedra lectures and will be introduced, in the lab sessions, to the software tools required to complete the homework assignments.
In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. In the project, students propose and execute meaningful analyses of a real-world dataset, which will require creativity and the application of the tools encountered in the course. The outcome of this team effort will be a project portfolio that will be made public (and available as open source).
At the end of the semester, students will take a 3-hour final exam in a classroom with their own computer, where they will be asked to complete a data analysis pipeline (both with code and extensive comments) on a dataset they have never worked with before.

Keywords

data science, data analysis, data mining, machine learning

Learning Prerequisites

Required courses

The student must have passed an introduction to databases course, OR a course in probability & statistics, OR two separate courses that include programming projects. Programming skills are required (in class we will use mostly Python).

Recommended courses

- CS-423 Distributed Information Systems
- CS-433 Machine Learning

Important concepts to start the course

programming, algorithms, probability and statistics, databases

Learning Outcomes

By the end of the course, the student must be able to:

- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the Data Science pipeline
- Perform data acquisition (data formats, dataset fusion, Web scrapers, REST APIs, open data, big data platforms, etc.)
- Perform data wrangling (fixing missing and incorrect data, data reconciliation, data quality assessments, etc.)
- Perform data interpretation (statistics, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline
- Perform data interpretation (statistics, correlation vs. causality, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
- Construct a coherent understanding of the techniques and software tools required to perform the fundamental steps of the data science pipeline

Transversal skills

- Give feedback (critique) in an appropriate fashion.
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

- Physical in-class recitations and lab sessions
- Homework assignments
- Course project

Expected student activities

Students are expected to:

- Attend the lectures and lab sessions
- Complete 2-3 homework assignments
- Conduct the class project
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- Homework
- Project
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

- http://ada.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-401

EE-554

Automatic speech processing

Magimai Doss Mathew

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The goal of this course is to provide the students with the main formalisms, models and algorithms required for the implementation of advanced speech processing applications (involving, among others, speech coding, speech analysis/synthesis, and speech recognition, speaker recognition).

Content

1. Introduction: Speech processing tasks, Speech science, language engineering applications.

2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming, speech representation learning

3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.

4. Speech Synthesis: speech synthesis models, concatenative synthesis, hidden Markov models (HMMs) based speech synthesis, Neural speech synthesis.

5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on HMMs, Neural networks-based speech recognition.

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, Text-dependent and Text-independent speaker verification, Gaussian mixture models-/HMM-based speaker verification, speaker embeddings based speaker verification, presentation attack detection (antispoofing).

7. Paralinguistic speech processing: fundamentals and applications (e.g., emotion recognition, pathological speech detection, depression detection), hand-crafted feature based approaches, neural approaches.

Keywords

speech processing, speech coding, speech analysis/synthesis, automatic speech recognition, speaker identification, text-to-speech

Learning Prerequisites

Required courses

Basis in linear algebra, signal processing (FFT), and statistics.

Important concepts to start the course

Basic knowledge in signal processing, linear algebra, statistics and stochastic processes. Basic knowledge in machine learning/statistical pattern recognition is not a must but would be helpful in .

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze speech signal properties
- Exploit those properties for speech coding, speech synthesis, speech recognition, speaker recognition and paralinguistic speech processing
- Formulate speech processing problems
- Choose appropriate methods for target speech processing tasks

Transversal skills

- Use a work methodology appropriate to the task.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Teaching methods

Lecture + lab exercises

Expected student activities

Attending courses and lab exercises. Read additional papers and continue lab exercises at home if necessary. Regulary answer list of questions for feedback.

Assessment methods

Written exam without notes

Resources

Bibliography

Fundamentals of Speech Recognition / Rabiner and Juang
Speech and Language Processing / Dan Jurafsky and Daniel Martin (2nd Edition)
Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Ressources en bibliothèque

- Speech and Language Processing / Dan Jurafsky and Daniel Martin
- Fundamentals of Speech Recognition / Rabiner and Juang
- Spoken language processing: A Guide to Theory, Algorithm and System Development / Xuedong Huang, Alex Acero and Hsiao-Wuen Hon
- Speech and Audio Signal Processing: Processing and Perception of Speech and Music / Ben Gold, Nelson Morgan and Dan Ellis
- Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / Björn Schuller and Anton Batliner

Moodle Link

- https://go.epfl.ch/EE-554

MICRO-452

Basics of mobile robotics

Mondada Francesco

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	1 weekly
Practical work	1 weekly
Project	2 weekly
Number of positions	

Summary

The course teaches the basics of autonomous mobile robots. Both hardware (energy, locomotion, sensors) and software (signal processing, control, localization, trajectory planning, high-level control) will be tackled. The students will apply the knowledge to program and control a real mobile robot.

Content

- Sensors
- Perception, feature extraction
- Modeling
- Markov localization: Bayesian filter, Monte Carlo localization, extended Kalman filter
- Navigation: path planning, obstacle avoidance
- Control architectures and robotic frameworks
- Locomotion principles and control
- Sustainability

Keywords

mobile robots, sensing, perception, localisation, navigation, locomotion.

Learning Prerequisites

Required courses

Introduction to automatic control (catching up possible with extra effort)
Introduction to signal processing

Recommended courses

Microinformatique (SMT)

Important concepts to start the course

Embedded system programming
Basics of automatic control

Learning Outcomes

By the end of the course, the student must be able to:

- Choose the right methods to design and control a mobile robot for a particular task.
- Integrate approriate methods for sensing, cognition and actuation
- Justify design choices for a robotic system
- Implement perception, localisation/navigation and control methods on a mobile robot
- Choose the right methods to design and control a mobile robot for a particular task.

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Set objectives and design an action plan to reach those objectives.
- Use a work methodology appropriate to the task.
- Assess progress against the plan, and adapt the plan as appropriate.
- Chair a meeting to achieve a particular agenda, maximising participation.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra, case studies, exercises, work on mobile robots, group project

Expected student activities

- weekly lectures
- studying provided additional materials
- attend case study discussions
- lab exercises with practical components
- project at the end of the semester

Assessment methods

Project during the semester (60% of the grade). The project takes place during the semester and the report and presentation are done before the end of the semester, following the specific planning given by the teacher at the beginning of the semester.
Written exam (40% of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Bibliography

Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004

Autonomous Robots: From Biological Inspiration to Implementation and Control G.A. Bekey, MIT Press, 2005
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005
Handbook of Robotics (chapter 35) B. Sicilian, and O. Khatib (Eds.), Springer, 2008
Elements of Robotics M. ben-Ari and F. Mondada, Spinger, 2017.
additional literature provided on Moodle

Ressources en bibliothèque

- Handbook of Robotics / Sicilian
- Elements of Robotics / Ben-Ari
- Probabilistic Robotics / Thrun
- Introduction to Autonomous Mobile Robots / Siegwart
- Autonomous Robots / Bekey

Notes/Handbook

Lecture slides are continously provided on Moodle during the course.
Introduction to Autonomous Mobile Robots R. Siegwart, and I. Nourbakhsh, MIT Press, 2004
Probabilistic Robotics S. Thrun, W. Burgard and D. Fox, MIT Press, 2005

Moodle Link

- https://go.epfl.ch/MICRO-452

BIO-410

Bioimage informatics

Sage Daniel, Seitz Arne

Cursus	Sem.	Type
Biomedical technologies minor	E	Opt.
Biotechnology minor	E	Opt.
Computer science	MA2, MA4	Opt.
Electrical Engineering		Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The course provides a comprehensive overview of methods, algorithms, and computer tools used in bioimage analysis. It exposes fundamental concepts and practical computer solutions to extract quantitative information from multidimensional images, both using engineering methods and deep learning.

Content

To investigate biological processes, bioimage informatics emerges as a growing field on the interface between microscopy, signal-processing, and computer science. The recent microscopes are producing large volumes of high-resolution multidimensional data (up to 5D). Therefore, algorithms and software tools are needed to automatically extract quantitative data from these images.
The course gives the theoretical concepts and practical aspects of the most common image reconstruction and image analysis techniques. It explains how to code algorithms and to deploy software tools to build an automatic analysis workflow using the most common used software, Fiji/ImageJ (Java-based) and Jupyter Notebook (Python-based). The lecture is tailored to the needs of life sciences and driven by biological questions.
Addressed topics include (but not restricted to): presentation of microscopy modalities, digital images, multi-dimensional data (3D, time, multiple channels) manipulation, 3D image-processing algorithms, 5D visualization, image metrics, reconstruction, deconvolution, denoising, stitching, visual feature detection, segmentation, active contours, image analysis workflow, pixel classification, machine learning, deep learning for image analysis, large datasets, tracking of particles, and super-resolution localization microscopy.
The course is composed of lectures, workshops, practices, and a mini-project.

A personal laptop is recommended to run (open-source) image analysis software and to develop short scripts.

Keywords

Bioimage, microscopy, image processing, image reconstruction, image analysis, visualization, multidimensional data analysis, machine learning, deep learning

Learning Prerequisites

Required courses

• Baisc knowledge in programming

Important concepts to start the course

- Basic knowledge in signal or image processing
- Baisc knowledge in programming

Learning Outcomes

- Identify quality of images in life science and expectation of the analysis
- Define the fundamental concepts of the computational bioimaging methods
- Select appropriately and compare methods and tools for common bioimage analysis tasks
- Design implements and experiment algorithms to solve specific tasks
- Develop a workflow for customized application
- Assess / Evaluate strategies for image-based experiments in life science

Transversal skills

- Demonstrate the capacity for critical thinking
- Use a work methodology appropriate to the task.
- Use both general and domain specific IT resources and tools

Assessment methods

20% Homework, individual

- In the first half of the semester: 4 homeworks on computer (2 weeks)

40% Mini-project by groups of 2-3 students

- In the second half of the semester: Development of an image analysis tool for a real application in biology

40% End-term exam, individual

- written exam with handwritten notes

Resources

Moodle Link

- https://go.epfl.ch/BIO-410

EPFL

MGT-416 **Causal inference**

Kiyavash Negar

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Management, Technology and Entrepreneurship minor	E	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Students will learn the core concepts and techniques of network analysis with emphasis on causal inference. Theory and application will be balanced, with students working directly with network data throughout the course.

Content

- Introduction: What is causal inference?
- Review of Useful Probability concepts
- Random variable, predictors, divergences
- Introduction to Applications
- Computational neuroscience
- Financial markets
- Social networks
- Pearl Causality
- Causal Bayesian Networks (CBNs)
- Learning CBNS: Faithfulness and identifiably
- Algorithms
- Potential Outcome Model
- Counterfactuals and identification problems
- Graphical causal models
- Randomized Experiments
- Identification of causes in randomized experiments
- Effect modification
- Causality in Times Series
- Granger causality
- More general linear predictors
- Beyond linear models and Granger causality
- Directed information graphs
- Efficient algorithms
- Concrete Applications
- Computational neuroscience
- Financial markets
- Social networks

Keywords

Causality, structure learning, network inference

Learning Prerequisites

Required courses

This course attempts to be as self contained as possible, but it does approach the topic from a quantitative point of view and, as such, students should be comfortable with the basics of (i.e. have taken at least one course in) the following topics before enrolling:
- Statistics
- Probability Theory
- Linear Algebra
- Calculus (integral and differential)
- Programing in Pythor and Matlab
As course work will be largely computational, experience with at least one programming language is also required.

Recommended courses

Knowledge of probability and calculus as well as programming is a must.

Learning Outcomes

By the end of the course, the student must be able to:

- Identify situations in which a problem/data can be thought of as a network.
- Analyze data appropriately using a variety of network causal inference techniques
- Interpret the results of applying causal discovery or inference algorithms

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

In class with supporing problem solving sessions.

Expected student activities

TA problem solving sessions, homework, exams, projects

Assessment methods

Regular individual assignments: 30%
Midterm project: 30%
Final project: 40%

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

course notes

Moodle Link

- https://go.epfl.ch/MGT-416

EPFL

MATH-352 **Causal thinking**

Stensrud Mats Julius

Cursus	Sem.	Type
Chemistry	BA5	Opt.
Computational and Quantitative Biology		Obl.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

This course will give a unified presentation of modern methods for causal inference. We focus on concepts, and we will present examples and ideas from various scientific disciplines, including medicine, computer science, engineering, economics and epidemiology.

Content

Association vs. causation
Definitions of causal effects
- Causal models
- Counterfactuals and potential outcomes
- Individual level causal effects vs. average causal effects
- Population causal effects
Study design
- Randomisation and experiments
- Observational studies
Causal graphs
- Causal Directed Acyclic Graphs
- Single World Intervention Graphs
Identification of causal effects
- Identifiability assumptions
- SWIGs
Causal mechanisms
- Mediation and path specific effects
- Instrumental variables
Applications
- Medical interventions, including pharmaceuticals
- Experiments in technology industry and engineering
- Experiments in life sciences
- Causal effects and mechanisms in the social sciences.
Estimation of causal effects
- Estimation using classical statistical models
- Estimation using machine learning

Keywords

Causality; Causal inference; Randomisation; Design of experiments; Observational studies; Causal Graphs

Learning Prerequisites

Required courses

The course is intended for students from a range of different disciplines, including computer science, engineering, life science and physics. The students are expected to know the basics of statistical theory and probability theory (such as the second year courses in probability and statistics for engineers).

Recommended courses

Courses in statistical inference.

Important concepts to start the course

Familiarity with basic concepts in probability and statistics.

Learning Outcomes

By the end of the course, the student must be able to:

- Design experiments that can answer causal questions.
- Describe the fundamental theory of causal models.
- Critique assess causal assumptions and axioms.
- Distinguish between interpretation, identification and estimation.
- Describe when and how causal effects can be identified and estimated from non- experimental data.
- Estimate causal parameters from observational data

Teaching methods

Classroom lectures, where I will use a digital blackboard and slides.

Assessment methods

Final written exam. 1-2 graded homeworks.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Hernan, M.A. and Robins, J.M., 2020. Causal inference: What if?
Imbens, G.W. and Rubin, D.B., 2015. Causal inference in statistics, social, and biomedical sciences. Cambridge University Press.
Pearl, J., 2009. Causality. Cambridge university press.

Ressources en bibliothèque

- Causal inference in statistics, social, and biomedical sciences / Imbens
- Causality / Pearl
- Causal Inference / Hernan

Moodle Link

- https://go.epfl.ch/MATH-352

BIO-105 Cellular biology and biochemistry for engineers

Zufferey Romain

Cursus	Sem.	Type
Biomedical technologies minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Minor in life sciences engineering	H	Opt.
Physics of living systems minor	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Basic course in biochemistry as well as cellular and molecular biology for non-life science students enrolling at the Master or PhD thesis level from various engineering disciplines. It reviews essential notions necessary for a training in biology-related engineering fields.

Content

The course gives basic knowledge on various phenoma taking place within a cell, and among cells within tissues and organs. The course gives an integrated view of various molecular mechanisms (rather in the second half of the class). It should therefore allow engineering students involved in future projects touching on biomedical problems to better integrate the constraints of a biological system and to enable them to communicate with specialists in both fields.
This course is not available to students who had already taken basic cell biology or biochemistry classes during their Bachelor studies at EPFL or elsewhere. This applies for example to the course BIO-109 "Introduction to Life Sciences for Information Sciences" and MSE 212 "Biology for engineers"

Keywords

The course contains chapters on the following subjects:
1.Cells and Organs
2.Chemical components of cells
3.Proteins, Enzymes
4.Energy, Metabolism
5.DNA, Chromosomes, Replication
6.Gene expression
7.Recombinant techniques
8.Membrane and Transport
9.Intracellular trafficking
10.Cytoskeleton
11.Cell division, Mitosis
12.Genetics, Meiosis
13.Cell communication, Signaling
14.Tissue, Tissue regeneration

Learning Prerequisites

Required courses

Bachelor degree in engineering or other non-life science discipline

Recommended courses

Some basic knowledge in chemistry can help, but not required

Important concepts to start the course

Curiosity about how biological systems work, willingness to acquire a certain amount of facts and details necessary to understand and discuss the various molecular mechanisms present in cells or related to modern biology

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the basic components and functions found in cells
- Draw schemes explaining essential cellular phenomena
- Explain which are the important metabolic pathways
- Translate information from genetic code
- Verify statements about specific cellular mechanisms
- Integrate knowledge from different cellular mechanisms

Transversal skills

- Access and evaluate appropriate sources of information.

Teaching methods

2 hours of ex cathedra-type of lecture
2 hours of exercises: the instructor gives out appr. 10 questions out (through Moodle and in the beginning of the session). The questions have different formats, and can in some cases just retrieve the acquired facts, in others have a more integrative problem-based learning approach.

Expected student activities

- review regularly the presented lectures.
- participate actively in the exercise sessions when the questions and problems are discussed altogether

Assessment methods

- a written exam at the winter exam session

Supervision

Office hours	Yes
Assistants	Yes
Forum	No
Others	- the teacher can always be reached through Email or phone to fix a one-to-one discussion about specific subjects

Resources

Bibliography

The lecture is aligned to selected chapters in the following book (recommended although not required): "Essential Cell Bioogy" by B Alberts et al. , 3rd edition, Garland Science Taylor & Francis Group

Notes/Handbook

- Essential Cell Biology / Alberts

Moodle Link

- https://go.epfl.ch/BIO-105

CS-524 Computational complexity

Göös Mika

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

In computational complexity we study the computational resources needed to solve problems and understand the relation between different types of computation. This course advances the students knowledge of computational complexity, and develop an understanding of fundamental open questions.

Content

- Complexity classes (time, space, nondeterminism)
- Space complexity (Logspace, L vs NL)
- Boolean circuits and nonuniform computation
- Power of randomness
- Lower bounds for concrete models of computation: Decision trees, communication protocols, propositional proofs.

Keywords

theoretical computer science
computational complexity

Learning Prerequisites

Recommended courses

Theory of computation (CS-251)
Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Demonstrate an understanding of computational complexity and the P vs NP problem
- Formalize and analyze abstractions of complex scenarios/problems
- Express a good understanding of different concepts of proofs
- Prove statements that are similar to those taught in the course
- Use and understand the role of randomness in computation

- Illustrate a basic understanding of probabilistically checkable proofs and their characterization of the class NP (the PCP-Theorem)
- Explain recent exciting developments in theoretical computer science
- Compare different models of computation

Transversal skills

- Demonstrate the capacity for critical thinking
- Summarize an article or a technical report.

Teaching methods

Lecturing and exercises

Expected student activities

Actively attending lectures and exercise sessions. Also homeworks and exam.

Assessment methods

Three homeworks and final exam

Resources

Bibliography

Sanjeev Arora and Boaz Barak: *Computational Complexity: A Modern Approach, Cambridge University Press.*
Stasys Jukna: *Boolean Function Complexity, Springer*

Ressources en bibliothèque

- Computational Complexity: A Modern Approach / Arora
- Boolean Function Complexity / Jukna

Moodle Link

- https://go.epfl.ch/CS-524

NX-465

Computational neurosciences: neuronal dynamics

Gerstner Wulfram

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
Biomedical technologies minor	E	Opt.
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Neuroscience		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

In this course we study mathematical models of neurons and neuronal networks in the context of biology and establish links to models of cognition. The focus is on brain dynamics approximated by deterministic or stochastic differential equations.

Content

I. Models of single neurons
1. Introduction: brain, computers, and a first simple neuron model
2. Models on the level of ion current (Hodgkin-Huxley model)
3./4. Two-dimensional models and phase space analysis

II. Neuronal Dynamics of Cognition
5. Associative Memory and Attractor Dynamics (Hopfield Model)
6. Neuronal Populations and mean-field methods
7. Continuum models and perception
8. Competition and models of Decision making

III. Noise and the neural code
9. Noise and variability of spike trains (point processes, renewal process, interval distribution)
10: Variance of membrane potentials and Spike Response Models
11. Population dynamics: Fokker-Planck equation
12. Fitting Neural Models to Data
13. Neural Manifolds and low-dimensional dynamics
14. Summary

Keywords

neural networks, neuronal dynamics, computational neuroscience, mathematical modeling in biology, applied mathematics, brain, cognition, neurons, memory, learning, plasticity

Learning Prerequisites

Required courses

undergraduate math at the level of electrical engineering or physics majors
undergraduate physics.

Recommended courses

Analysis I-III, linear algebra, probability and statistics
For SSV students: Dynamical Systems Theory for Engineers or "Mathematical and Computational Models in Biology"

Important concepts to start the course

Differential equations, Linear equations,

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze two-dimensional models in the phase plane
- Solve linear one-dimensional differential equations
- Develop a simplified model by separation of time scales
- Analyze connected networks in the mean-field limit
- Predict outcome of dynamics
- Prove stability and convergence
- Describe neuronal phenomena
- Test model concepts in simulations

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Write a scientific or technical report.

Teaching methods

- Video lectures, combined with classroom teaching, classroom discussions, exercises and miniproject. One of the two exercise hours is integrated into the lectures.
- Short mooc-style videos are available as support
- Textbook available as support

Expected student activities

-participate in ALL in-class exercises.
- do all homework exercises (paper-and-pencil)
- study video lectures if you miss a class
- study suggested textbook sections for in-depth understanding of material
- submit miniprojects

Assessment methods

Written exam (70%) & miniproject (30%)
The miniproject is done in teams of 2 students.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	The teacher is available during the breaks of the class. Some exercises are integrated in class in the presence of the teacher and the teaching assistants.

Resources

Bibliography

Gerstner, Kistler, Naud, Pansinski : Neuronal Dynamics, Cambridge Univ. Press 2014

Ressources en bibliothèque

- Neuronal dynamics: from single neurons to networks and models of cognition / Wulfram Gerstner, Werner M. Kistler, Richard Naud, Liam Paninski

Websites

- https://neuronaldynamics.epfl.ch/
- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

Moodle Link

- https://go.epfl.ch/NX-465

Videos

- https://lcnwww.epfl.ch/gerstner/NeuronalDynamics-MOOCall.html

CS-413 Computational photography

Süsstrunk Sabine

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The students will gain the theoretical knowledge in computational photography, which allows recording and processing a richer visual experience than traditional digital imaging. They will also execute practical group projects to develop their own computational photography application.

Content

Computational photography is the art, science, and engineering of creating a great (still or moving) image. Information is recorded in space, time, across visible and invisible radiation and from other sources, and then post-processed to produce the final - visually pleasing - result.
Basics: Human vision system, Light and illumination, Geometric optics, Color science, Sensors, Digital camera systems.
Generalized illumination: Structured light, High dynamic range (HDR) imaging, Time-of-flight.
Generalized optics: Coded Image Sensing, Coded aperture, Focal stacks.
Generalized sensing: Low light imaging, Depth imaging, Plenoptic imaging, Light field cameras.
Generalized processing: Super-resolution, In-painting, Compositing, Photomontages, Panoramas, HDR imaging, Multi-wavelength imaging, Dynamic imaging.
Generalized display: Stereoscopic displays, HDR displays, 3D displays, Mobile displays.
Deep Learning for image resoration and image enhancement.

Keywords

Computational Photography, Coded Image Sensing, Non-classical image capture, Multi-Image & Sensor Fusion, Mobile Imaging, Machine Learning

Learning Prerequisites

Required courses

- A basic Signal Processing, Image Processing, and/or Computer Vision course.
- Linear Algebra.

Recommended courses

- Computer Vision.
- Signal Processing
- Machine Learning.

Important concepts to start the course

• Basic signal/image processing.
• Basic computer vision.
• Basic programming (Python, iOS, Android).

Learning Outcomes

By the end of the course, the student must be able to:

• Identify the main components of a computational photography system.
• Contextualise the main trends in computational optics, sensing, processing, and displays.
• Create a computational photography application on a mobile platform.
• Design a computational photography solution to solve a particular imaging task.
• Assess / Evaluate hardware and software combinations for their imaging performance.
• Formulate computational photography challenges that still need to be resolved.
• Create a computational photography application.

Transversal skills

• Evaluate one's own performance in the team, receive and respond appropriately to feedback.
• Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

The course consists of 2 hours of lectures per week that will cover the theoretical basics. An additional 2 hours per week are dedicated to a group project designing, developing, and programming a compuational photography application on a mobile plateform (iOS, Android).

Expected student activities

The studens is expected to attend the class and actively participate in the practical group project, which requires coding on either Android or iOS plateform. The student is also required to read the assigned reading material (book chapters, scientific articles).

Assessment methods

The theoretical part will be evaluated with an oral exam at the end of the semester, and the practical part based on the students' group projects

Resources

Bibliography

• Selected book chapters
• Course notes (on moodle)
• Links to relevant scientific articles and on-line resources will be given on moodle.

Moodle Link

• https://go.epfl.ch/CS-413

CS-442

Computer vision

Fua Pascal

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Computer Vision aims at modeling the world from digital images acquired using video or infrared cameras, and other imaging sensors. We will focus on images acquired using digital cameras. We will introduce basic processing techniques and discuss their field of applicability.

Content

Introduction

- History of Computer Vision
- Human vs Machine Vision
- Image formation

Extracting 2D Features

- Contours
- Texture
- Regions

3D Shape Recovery

- From one single image
- From multiple images

Learning Outcomes

By the end of the course, the student must be able to:

- Choose relevant algorithms in specific situations
- Perform simple image-understanding tasks

Teaching methods

Ex cathedra lectures and programming exercises using Python.

Assessment methods

With continuous control

Resources

Bibliography

- R. Szeliski, Computer Vision: Computer Vision: Algorithms and Applications, 2010.
- A. Zisserman and R. Hartley, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

Ressources en bibliothèque

- Multiple View Geometry in Computer Vision / Zisserman
- Computer Vision: Algorithms and Applications / Szeliski

Moodle Link

- https://go.epfl.ch/CS-442

EPFL

COM-418 Computers and music

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

In this class we will explore some of the fundamental ways in which the pervasiveness of digital devices has completely revolutionized the world of music in the last 40 years, both from the point of view of production and recording, and from the point of view of listening and distribution.

Content

- review of digital signal processing: discrete-time signals, spectral analysis, digital filters
- audio measurement standards; A/D and D/A converters; oversampling; sigma-delta
- audio compression; the MP3 standard
- digital synthesizers: oscillators, FM synthesis, samplers
- fundamentals of time-frequency analysis; pitch shifting; time stretching; vocoder
- music production; equalization, compression, reverb
- notions of balancing and mastering; the MIDI and VST standards
- nonlinear system modeling
- deep learning in audio processing

Keywords

DSP, computer music, digital audio

Learning Prerequisites

Recommended courses

Signals and systems, Python, C++

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the fundamental techniques in digital audio recording and production
- Be able to avoid unwanted artifacts in sound recording and compression
- Recognize the typical acoustic footprint of classic synthesizers and audio effects

- Write working signal processing code to synthesize sounds and process audio
- Write code that interfaces to existing equipment via industry-standard protocols

Transversal skills

- Access and evaluate appropriate sources of information.
- Summarize an article or a technical report.
- Write a scientific or technical report.
- Demonstrate a capacity for creativity.

Teaching methods

lectures

Expected student activities

- Attending lectures
- Writing code samples
- Solving exercises
- Read technical papers

Assessment methods

Mini-projects and/or final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/COM-418

CS-453

Concurrent computing

Guerraoui Rachid

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

With the advent of modern architectures, it becomes crucial to master the underlying algorithmics of concurrency. The objective of this course is to study the foundations of concurrent algorithms and in particular the techniques that enable the construction of robust such algorithms.

Content

Model of a parallel system
Multicore and multiprocessors architecture

Processes and objects

Safety and liveness

Parallel programming
Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

Register Implementations
Safe, regular and atomic registers
Counters General and limited operations
Atomic counters and snapshots

Hierarchy of objects
The FLP impossibility
The consensus number
Universal constructions

Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Anonymous computing

Fault-tolerant shared-memory computing

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses

ICC, Operating systems

Recommended courses

This course is complementary to the Distributed Algorithms course

Important concepts to start the course

Processes, threads, datas structures

Learning Outcomes

By the end of the course, the student must be able to:

- Reason in a precise manner about concurrency
- Design a concurrent algorithm
- Prove a concurrent algorithm
- Implement a concurrent system

Teaching methods

Lectures, exercises and practical work

Expected student activities

Final exam
Project

Assessment methods

Final exam (theory) and project (practice)

Resources

Notes/Handbook

Algorithms for Concurrent Systems, R. Guerraoui and P. Kouznetsov

Moodle Link

- https://go.epfl.ch/CS-453

COM-401

Cryptography and security

Vaudenay Serge

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

This course introduces the basics of cryptography. We review several types of cryptographic primitives, when it is safe to use them and how to select the appropriate security parameters. We detail how they work and sketch how they can be implemented.

Content

1. **Ancient cryptography**: Vigenère, Enigma, Vernam cipher, Shannon theory
2. **Diffie-Hellman cryptography**: algebra, Diffie-Hellman, ElGamal
3. **RSA cryptography**: number theory, RSA, factoring
4. **Elliptic curve cryptography**: elliptic curves over a finite field, ECDH, ECIES, pairing
5. **Symmetric encryption**: block ciphers, stream ciphers, exhaustive search
6. **Integrity and authentication**: hashing, MAC, birthday paradox
7. **Public-key cryptography**: cryptosystem, digital signature, post-quantum cryptography
8. **Trust establishment**: password-based cryptography, secure communication, trust setups
9. **Case studies**: WiFI, bitcoin, mobile telephony, WhatsApp, EMV, Bluetooth, biometric passport, TLS

Keywords

cryptography, encryption, secure communication

Learning Prerequisites

Required courses

MATH-310 Algebra
MATH-232 Probability and statistics for IC
CS-250 Algorithms I

Recommended courses

COM-301 Computer security and privacy

Important concepts to start the course

• Mathematical reasoning
• Probabilities

- Algebra, arithmetics
- Algorithmics

Learning Outcomes

By the end of the course, the student must be able to:

- Choose the appropriate cryptographic primitive in a security infrastructure
- Judge the strength of existing standards
- Assess / Evaluate the security based on key length
- Implement algorithms manipulating big numbers and use number theory
- Use algebra and probability theory to analyze cryptographic algorithms
- Identify the techniques to secure the communication and establish trust

Teaching methods

ex-cathedra

Expected student activities

- active participation during the course
- take notes during the course
- do the exercises during the exercise sessions
- complete the regular tests and homework
- read the material from the course
- self-train using the provided material
- do the midterm exam and final exam

Assessment methods

Mandatory continuous evaluation:

- homework (30%)
- regular graded tests (30%)
- midterm exam (40%)

Final exam averaged (same weight) with the contiuous evaluation, but with final grade between final_exam-1 and final_exam+1.

Supervision

Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Bibliography

- Communication security: an introduction to cryptography. Serge Vaudenay. Springer 2004.
- A computational introduction to number theory and algebra. Victor Shoup. Cambridge University Press 2005.

Ressources en bibliothèque

- A Classical Introduction to Cryptography / Vaudenay
- A computational introduction to number theory and algebra / Shoup

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-401

Videos

- https://mediaspace.epfl.ch/channel/COM-401+Cryptography+and+security

Prerequisite for

- Advanced cryptography (COM-501)
- Student seminar: security protocols and applications (COM-506)

EPFL

COM-480 **Data visualization**

Vuillon Laurent Gilles Marie

Cursus	Sem.	Type
Computational and Quantitative Biology		Obl.
Computational biology minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Understanding why and how to present complex data interactively in an effective manner has become a crucial skill for any data scientist. In this course, you will learn how to design, judge, build and present your own interactive data visualizations.

Content

Tentative course schedule
Week 1: Introduction to Data visualization Web development
Week 2: Javascript
Week 3: More Javascript
Week 4: Data Data driven documents (D3.js)
Week 5: Interaction, filtering, aggregation (UI /UX). Advanced D3 / javascript libs
Week 6: Perception, cognition, color Marks and channels
Week 7: Designing visualizations (UI/UX) Project introduction Dos and don’ts for data-viz
Week 8: Maps (theory) Maps (practice)
Week 9: Text visualization
Week 10: Graphs
Week 11: Tabular data viz Music viz
Week 12: Introduction to scientific visualisation
Week 13: Storytelling with data / data journalism Creative coding
Week 14: Wrap-Up

Keywords

Data viz, visualization, data science

Learning Prerequisites

Required courses

CS-250 Algorithms I (BA)
CS-401 Applied data analysis (MA)

Recommended courses

CS-486 Interaction design (MA)
CS-214 Software construction (BA)

Important concepts to start the course

Being autonomous is a prerequisite, we don't offer office hours and we won't have enough teaching assistants (you've been warned!).
Knowledge of one of the following progrmaming language such as C++, Python, Scala.
Familiarity with web-development (you already have a blog, host a webiste). Experience with HTML5, Javascript is a strong plus for the course.

Learning Outcomes

By the end of the course, the student must be able to:

- Judge visualization in a critical manner and suggest improvements.
- Design and implement visualizations from the idea to the final product according to human perception and cognition
- Know the common data-viz techniques for each data domain (multivariate data, networks, texts, cartography, etc) with their technical limitations
- Create interactive visualizations int he browser using HTM5 and Javascript

Transversal skills

- Communicate effectively, being understood, including across different languages and cultures.
- Negotiate effectively within the group.
- Resolve conflicts in ways that are productive for the task and the people concerned.

Teaching methods

Ex cathedra lectures, exercises, and group projects

Expected student activities

- Follow lectures
- Read lectures notes and textbooks
- Create an advanced data-viz in groups of 3.
- Answer questions assessing the evolution of the project.
- Create a 2min screencast presentation of the viz.
- Create a process book for the final data viz.

Assessment methods

- Data-viz (35%)
- Technical implementation (15%)
- Website, presentation, screencast (25%)
- Process book (25%)

Resources

Bibliography

Visualization Analysis and Design by Tamara Munzner, CRC Press (2014). Free online version at EPFL.
Interactive Data Visualization for the Web by Scott Murray O'Reilly (2013) - D3 - Free online version.
The Truthful Art: Data, Charts, and Maps for Communication by Cairo, Alberto. Royaume-Uni, New Riders, (2016).
Data Visualisation: A Handbook for Data Driven Design by Kirk, Andy. Royaume-Uni, SAGE Publications, (2019).

Ressources en bibliothèque

- Data Visualisation / Kirk
- Visualization Analysis and Design / Munzner
- The Truthful Art / Cairo
- Interactive Data Visualization for the Web / Murray [2 ed. 2017]

Notes/Handbook

Lecture notes

Moodle Link

- https://go.epfl.ch/COM-480

EPFL

CS-438

Decentralized systems engineering

Borsò-Tan Pierluca, Ford Bryan Alexander

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

A decentralized system is one that works when no single party is in charge or fully trusted. This course teaches decentralized systems principles while guiding students through the engineering of their own decentralized system featuring messaging, file sharing, encryption, and blockchain concepts.

Content

• Addressing, Forwarding, Routing. Peer-to-peer communication.

• Information gossip. UseNet: technical, security, and social lessons. Randomized rumor-mongering and anti-entropy algorithms.

• Trust and Reputation. Authorities, trust networks. Sybil attacks and defenses.

• Naming and search. Request flooding. Hierarchical directories, landmark routing. Self-certifying identities. Distributed hash tables.

• Distributed consensus, distributed ledgers (blockchains), and cryptocurrencies.

• Anonymous Communication. Onion routing, mix networks. Dining cryptographers. Voting, verifiable shuffles, homomorphic encryption. Anonymous disruption.

• Fireproofing Alexandria: Decentralized Storage. Replication. Parity, erasure coding. Renewal. Digital preservation.

• Content Distribution. Opportunistic caching (FreeNet). Content integrity: hash trees, hash file systems. Convergent encryption. Swarming downloads: BitTorrent. Free-riding, incentives.

• Gaining perspective. Spam, malicious content. Review/moderation and reputation systems. Leveraging social networks (Peerspective). Balancing local and global viewpoints.

• Decentralized Collaboration. Network file systems, version management. Consistency.

• Consistency Models. Disconnected operation, eventual consistency, conflict resolution.

• Distributed Consensus. Paxos. Accountability (PeerReview). Byzantine fault tolerance.

• Mobile Code. Smart contract systems. Privacy: trusted computing, fully homomorphic encryption. Decentralized virtual organizations.

• Securing Decentralized Systems: threat modelling, use of threshold cryptosystems.

• Going into production: Quality assurance, chaos engineering and operations.

Learning Prerequisites

Required courses

• CS-202 Computer systems or COM-208 Computer networks

- COM-301 Computer security and privacy

Learning Outcomes

By the end of the course, the student must be able to:

- Implement decentralized systems via hands-on coding, debugging, and testing
- Design effective testing strategies for distributed and decentralized systems
- Design practical distributed and decentralized systems

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Assess progress against the plan, and adapt the plan as appropriate.
- Take account of the social and human dimensions of the engineering profession.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Lectures: The course's lectures will present and discuss challenges, known techniques, and open questions in decentralized system design and implementation. Lectures will often be driven by examination of real decentralized systems with various purposes in widespread use the past or present, such as UseNet, IRC, FreeNet, Tor, BitTorrent, and Bitcoin. Throughout the course we will explore fundamental security and usability challenges such as decentralized identification and authentication, denial-of-service and Sybil attacks, and maintenance of decentralized structures undergoing rapid changes (churn).
Labs: During the semester, students will develop a small but usable peer-to-peer communication application that reflects a few of the important design principles and techniques to be explored in the course, such as gossip, distributed hash tables, consensus algorithms, and cryptocurrencies. The labs will designed so that solutions can initially be tested individually on private, virtual networks running on one machine, then tested collectively by attempting to make different students' solutions interoperate on a real network.

Warning: This course is extremely programming-intensive. Students should be strong and confident in their programming skills in general, and be willing to spend substantial time outside of class debugging difficult distributed concurrency bugs and other challenges. TAs will be available to help at the exercise sessions, but *they will not solve your problems or debug your code for you*.

Expected student activities

Students will be expected to attend lectures to understand the concepts needed for the course, but the main workload will be actual hands-on programming assignments, which the students will perform on their own during the first part of the course and in teams during the final project-oriented part of the course.

Assessment methods

- Programming assignment grading (evaluating function, performance, correctness and implementation quality): 40%
- Team project grading (evaluating for scope, implementation quality, testing, evaluation, documentation and report): 30%
- Written exam: 30%

Resources

Virtual desktop infrastructure (VDI)
No

Moodle Link

- https://go.epfl.ch/CS-438

EE-559

Deep learning

Cavallaro Andrea

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Financial engineering	MA2, MA4	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in Quantum Science and Engineering	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	**150**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This course explores how to design reliable discriminative and generative neural networks, the ethics of data acquisition and model deployment, as well as modern multi-modal models.

Content

This course equips students with a comprehensive foundation of modern deep learning, enabling students to design and train discriminative and generative neural networks for a wide range of tasks. Topics include:

- Deep learning applications (natural language processing, computer vision, audio processing, biology, robotics, science), principles and regulations
- Loss functions, data and labels, data provenance
- Training models: gradients and initialization
- Generalization and performance
- Transformers
- Graph neural networks
- Generative adversarial networks
- Variational autoencoders
- Diffusion models
- Multi-modal models
- Interpretability, explanations, bias and fairness

Keywords

machine learning, neural networks, deep learning, python

Learning Prerequisites

Required courses

- Basics in probabilities and statistics
- Linear algebra
- Differential calculus
- Python programming

Recommended courses

- Basics in optimization
- Basics in algorithmic
- Basics in signal processing

Important concepts to start the course

Discrete and continuous distributions, normal density, law of large numbers, conditional probabilities, Bayes, PCA, vector, matrix operations, Euclidean spaces, Jacobian, Hessian, chain rule, notion of minima, gradient descent, computational costs, Fourier transform, convolution.

Learning Outcomes

By the end of the course, the student must be able to:

- Interpret the performance of a deep learning model
- Analyze the limitations of a deep learning model
- Justify the choices for training and testing a deep learning model
- Propose new solutions for a given application

Transversal skills

- Respect relevant legal guidelines and ethical codes for the profession.
- Take account of the social and human dimensions of the engineering profession.
- Design and present a poster.
- Make an oral presentation.
- Demonstrate the capacity for critical thinking

Teaching methods

Ex-cathedra lectures, class discussion, exercises (using python), group project.

Expected student activities

Attendance to lectures, participation in discussions, completing exercises, completing a project, reading written material (scientific papers and books).

Assessment methods

Excercises and group project.

Resources

Références suggérées par la bibliothèque

- Deep learning, Goodfellow, MIT Press, 2016

Notes/Handbook

Not mandatory: http://www.deeplearningbook.org/

Moodle Link

- https://go.epfl.ch/EE-559

CS-502 **Deep learning in biomedicine**

Cursus	Sem.	Type
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Minor in life sciences engineering	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

Deep learning offers potential to transform biomedical research. In this course, we will cover recent deep learning methods and learn how to apply these methods to problems in biomedical domain.

Content

The goal of this course is to cover recent deep learning methods and demonstrate how they can be applied to biomedical data. The course will cover ongoing advances in deep learning research for different input data types (e.g., convolutional neural networks for images, graph convolutional neural networks for graph structured data, transformers for sequence data). We will start with a standard supervised learning setting and then cover the ongoing developments in methodologies that allow us to learn using scarcely labeled datasets by transferring knowledge across tasks (e.g., transfer learning, meta-learning). These settings have particular importance in the biomedical domain in which it is often very difficult to obtain labeled datasets. Recent papers from the literature that apply these methods to problems in biomedicine will be presented and discussed.
In assignments, students will work with popular deep learning software frameworks. They will be evaluated on their ability to understand and implement the methods learned in a class. In the project, students will choose a real-world problem in the biomedical domain and develop a solution for the problem of their choice. They will be evaluated on the ability to propose and develop a suitable model to solve the task, propose suitable evaluation, provide analysis and extract insights from the developed models, write a project report and present project results.
This course is of interest to MS/PhD students interested in recent deep learning methods and their applications to real-world problems in the biomedical domain.

Learning Prerequisites

Required courses

CS-433 Machine learning

Recommended courses

CS-233 Introduction to machine learning

Important concepts to start the course

- Python programming

- Probability and statistics
- Linear Algebra
- Machine learning

Learning Outcomes

By the end of the course, the student must be able to:

- Understand and implement deep learning methods covered in the course
- Understand benefits and shortcomings of the methods covered in the course
- Understand common problems in the biomedical domain and know which methods are suitable for solving these problems
- Review academic research papers and understand their contributions according to concepts covered in the course
- Complete a project that applies learned algorithms to a real-world problem in the biomedical domain

Teaching methods

- Lectures
- Paper reading
- Course project

Expected student activities

- Attend lectures and participate in class
- Complete homework assignments
- Complete a deep learning project in a group. This includes preparing a project proposal, implementing the method, submitting final project report and presenting project results

Assessment methods

- Assignments during the semester (50%)
- Project (50%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Goodfello, Bengio, Courville. Deep Learning. MIT Press (2016)

Ressources en bibliothèque

- Deep Learning / Goodfellow

Moodle Link

- https://go.epfl.ch/CS-502

CS-456

Deep reinforcement learning

Gulcehre Caglar

Cursus	Sem.	Type
Computational biology minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Obl.
Financial engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course provides an overview and introduces modern methods for reinforcement learning (RL.) The course starts with the fundamentals of RL, such as Q-learning, and delves into commonly used approaches, like PPO and DQN. The course will introduce students to practical applications of RL.

Content

- *Introduction and Overview (What is RL?)*
- *An overview of neural networks and deep learning approaches*
- *Deep learning frameworks*
- *Supervised learning of behaviors (behavior cloning)*
- *Value function methods and related theory*
- *Policy gradient methods and related theory*
- *Actor-Critic Algorithms (A2C, A3C)*
- *Deep RL with Q functions (DQN, R2D2)*
- *Deep Policy Gradient and Optimization methods (PPO, TRPO, Impala, MPO)*
- *Model-based RL and Planning (Alphago, Alphazero, Dreamer)*
- *Exploration and credit assignment in Deep RL*
- *Offline RL (BVE, CQL, CRR, ...)*
- *Deep Imitation learning and Learning from demonstrations (DAGGER, DQFD, R2D3, Learning from play, Third*

person imitation)

• *RL from human feedback and alignment (InstructGPT, DPO, ReST, etc.)*

• *Advanced continuous control approaches (DDPG, D4PG, SAC)*

• *A selection of extra topics from:*

- MPO, IMPALA
- Distributional RL
- Multi-agent RL (Centralized Training, Decentralized Execution)

Keywords

Deep learning, reinforcement learning, TD learning, SARSA, Actor-Critic Networks, policy gradients, alphago, alphastar, planning, alignment, RLHF, PPO

Learning Prerequisites

Required courses

• Analysis I, II
• Linear Algebra
• Probability and statistics (MATH-232)
• Algorithms I (CS-250)

Recommended courses

• Introduction to machine learning (CS-233)
• Machine learning (CS-433)

Important concepts to start the course

• *Regularization in machine learning,*
• *Gradient descent. Stochastic gradient descent.*
• *Expectation, statistics*
• *Linear algebra and probabilities*
• *programming*

Learning Outcomes

By the end of the course, the student must be able to:

• Apply Understand and define basic problems and tasks in reinforcement learning (like Markov decision process, model-based and model-free RL, on-policy vs off-policy RL)

• Assess / Evaluate Formulate a real-world problem as an RL setting to apply the approaches taught in the class.

• Elaborate Implement standard deep RL algorithms.

• Judge Understand the failure modes of these models and learning algorithms.

• Propose Read and review academic papers to understand their contributions and learn how to evaluate them critically.

• Apply Students gain the skills and knowledge necessary to tackle complex problems in autonomous robotics, game-playing, and other domains through lectures, hands-on coding exercises, practical applications, and course projects.

Transversal skills

- Continue to work through difficulties or initial failure to find optimal solutions.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Manage priorities.

Teaching methods

- Lectures
- Lab sessions
- Individual course projects
- Paper reading
- Group projects

Expected student activities

- Work on miniproject
- Solve all exercises
- Attend all lectures and take notes during lecture, participate in quizzes.
- If you cannot attend a lecture, then you must read the recommended book chapters
- Work on a project

Assessment methods

- Written final exam (25%)
- Assignments (25%)
- Course project (50%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	TAs are available during exercise sessions. Every week one of the exercises is run as 'integrated exercise' during the lecture.

Resources

Bibliography

• Textbook: Reinforcement Learning by Sutton and Barto (MIT Press). Pdfs of the preprint version of the book are availble online

Ressources en bibliothèque

• Reinforcement Learning / Sutton

Moodle Link

• https://go.epfl.ch/CS-456

CS-472 Design technologies for integrated systems

De Micheli Giovanni

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
MNIS	MA3	Obl.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Labs	2 weekly
Number of positions	

Summary

Hardware compilation is the process of transforming specialized hardware description languages into circuit descriptions, which are iteratively refined, detailed and optimized. The course presents algorithms, tools and methods for hardware compilation and logic synthesis.

Content

The course will present the most outstanding features of hardware compilation, as well as the techniques for optimizing logic representations and networks. The course gives a novel, uptodate view of digital circuit design. Practical sessions will teach students the use of current design tools.Syllabus1) Modeling languages and specification formalisms;2) High-level synthesis and optimization methods (scheduling, binding, data-path and control synthesis);3) Representation and optimization of combinational logic functions (encoding problems, binary decision diagrams);4) Representation and optimization of multiple-level networks (algebraic and Boolean methods, "don't care" set computation, timing verification and optimization);5) Modeling and optimization of sequential functions and networks (retiming);6) Semicustom libraries and library binding.

Keywords

Hardware, VLSI, Synthesis, Optimization, Algorithms

Learning Prerequisites

Required courses

No specific course

Recommended courses

Good knowledge of digital design, algorithm design and programming.

Important concepts to start the course

Good knowledge of digital design, algorithm design and programming.

Learning Outcomes

By the end of the course, the student must be able to:

- Recognize important problems in digital design
- Examine and evaluate available design tools and methods
- Decide upon a design tool flow to perform a digital design

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.

Assessment methods

Continuous control :
Homework : 30 %, Project 10 %, Midterm test : 25 %,
End term test : 35 %

Resources

Bibliography

G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw'Hill.

Ressources en bibliothèque

- Synthesis and Optimization of Digital Circuits / De Micheli

Moodle Link

- https://go.epfl.ch/CS-472

CS-411

Digital education

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Winter, Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

This course addresses the relationship between specific technological features and the learners' cognitive processes. It also covers the methods and results of empirical studies: do student actually learn due to technologies? I will be given twice in 2025 (Spring + Fall)

Content

- *Learning theories and learning processes.*
- *Types of learning technlogies*
- *Instructional design: methods, patterns and principles.*
- *On-line education.*
- *Effectiveness of learning technologies.*
- *Methods for empirical research.*
- *Computational thinkiing skills*
- *Maker spaces*

Keywords

learning, pedagogy, teaching, online education, maker spaces

Learning Outcomes

By the end of the course, the student must be able to:

- Describe the learning processes triggered by a technology-based activity
- Explain how a technology feature influences learning processes
- Elaborate a study that measures the learning effects of a digital environment
- Select appropriately a learning technology given the target audience and the expected learning outcomes

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Take account of the social and human dimensions of the engineering profession.

- Write a scientific or technical report.

Teaching methods

The course will combine participatory lectures with a project on designing a learning environment, using it in a controlled experiment, analysing the learning effects with statistical methods and writing a report.

Expected student activities

The project will include a few milestones to be delivered along the semester.

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-411

EPFL

CS-451 Distributed algorithms

Guerraoui Rachid

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	3 weekly
Number of positions	

Summary

Computing is nowadays distributed over several machines, in a local IP-like network, a cloud or a P2P network. Failures are common and computations need to proceed despite partial failures of machines or communication links. This course will study the foundations of reliable distributed computing.

Content

Reliable broadcast
Causal Broadcast
Total Order Broadcast
Consensus
Non-Blocking Atomic Commit
Group Membership, View Synchrony
Terminating Reliable Broadcast
Shared Memory in Message Passing Systems
Byzantine Fault Tolerance
Self Stabilization
Population protocols   (models of mobile networks)
Bitcoin, Blockchain
Distributed Machine Learning
Gossip

Keywords

Distributed algorithms, checkpointing, replication, consensus, atomic broadcast, ditributed transactions, atomic commitment, 2PC, Machine Learning

Learning Prerequisites

Required courses

Basics of Algorithms, networking and operating systems

Recommended courses

The lecture is orthogonal to the one on concurrent algorithms: it makes a lot of sense to take them in parallel.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose an appropriate abstraction to model a distributed computing problem
- Specify the abstraction
- Present and implement it
- Analyze its complexity
- Prove a distributed algorithm
- Implement a distributed system

Teaching methods

Ex cathedera
Lectures, exercises and practical work

Assessment methods

Final exam (theory)
Project (practice)

Resources

Ressources en bibliothèque

- Introduction to reliable and secure distributed programming / Cachin

Notes/Handbook

Reliable and Secure Distributed Programming
Springer Verlag
C. Cachin, R. Guerraoui, L. Rodrigues

Moodle Link

- https://go.epfl.ch/CS-451

CS-423 Distributed information systems

Aberer Karl

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course introduces the foundations of information retrieval, data mining and knowledge bases, which constitute the foundations of today's Web-based distributed information systems.

Content

Information Retrieval

1. Information Retrieval - Introduction
2. Text-Based Information Retrieval (Boolean, Vector space, probabilistic)
3. Inverted Files
4. Distributed Retrieval
5. Query Expansion
6. Embedding models (LSI, word2vec)
7. Link-Based Ranking

Mining Unstructured Data

1. Document Classification (knn, Naive Bayes, Fasttext, Transformer models)
2. Recommender Systems (collaborative filtering, matrix factorization)
3. Mining Social Graphs (modularity clustering, Girvan-Newman)

Knowledge Bases

1. Semantic Web
2. Keyphrase extraction
3. Named entity recognition
4. Information extraction
5. Taxonomy Induction
6. Entity Disambiguation
7. Label Propagation
8. Link Prediction

Learning Prerequisites

Recommended courses

Introductory courses to databases and machine learning are helpful, but not required.
Programming skills in Python are helpful, but not required.

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the main tasks performed by information systems, namely data, information and knowledge management
- Apply collaborative information management models, like crowd-sourcing, recommender systems, social networks
- Apply knowledge models, their representation through Web standards and algorithms for storing and processing semi-structured data
- Apply fundamental models and techniques of text retrieval and their use in Web search engines
- Apply main categories of data mining techniques, local rules, predictive and descriptive models, and master representative algorithms for each of the categories

Teaching methods

Ex cathedra + programming projects (Python)

Assessment methods

60% Continuous evaluations of projects with bonus system during the semester
40% Final written exam (180 min) during exam session

Resources

Moodle Link

- https://go.epfl.ch/CS-423

EPFL

ENG-466 **Distributed intelligent systems**

Martinoli Alcherio

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Energy Science and Technology	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Mechanical engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	Oral
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Project	1 weekly
Labs	2 weekly
Number of positions	**40**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

The goal of this course is to provide methods and tools for modeling distributed intelligent systems as well as designing and optimizing coordination strategies. The course is a well-balanced mixture of theory and practical activities.

Content

- Introduction to key concepts such as self-organization and tools used in the course
- Examples of natural, artificial and hybrid distributed intelligent systems
- Modeling methods: sub-microscopic, microscopic, macroscopic, multi-level; spatial and non-spatial; mean field, approximated and exact approaches
- Machine-learning methods: single- and multi-agent techniques; expensive optimization problems and noise resistance
- Coordination strategies and distributed control: direct and indirect schemes; algorithms and methods; performance evaluation
- Application examples in distributed sensing and action

Keywords

Swarm intelligence, artificial intelligence, machine-learning, distributed robotics, swarm robotics, multi-robot systems, sensor networks, modeling, control, optimization

Learning Prerequisites

Required courses

Fundamentals in analysis, probability, and programming for both compiled and interpreted languages

Recommended courses

Basic knowledge in statistics, programming language used in the course (C, Matlab, Python), and signals and systems

Learning Outcomes

By the end of the course, the student must be able to:

- Design control algorithms
- Formulate a model at different level of abstraction for a distributed intelligent system
- Analyze a model of a distributed intelligent system
- Analyze a distributed coordination strategy/algorithm
- Design a distributed coordination strategy/algorithm
- Implement code for single robot and multi-robot systems
- Carry out systematic performance evaluation of a distributed intelligent system
- Apply modeling and design methods to specific problems requiring distributed sensing and action
- Optimize a controller or a set of possibly coordinated controllers using model-based or data-driven methods

Transversal skills

- Demonstrate a capacity for creativity.
- Access and evaluate appropriate sources of information.
- Collect data.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

Ex-cathedra lectures, assisted exercises, and course project in teams

Expected student activities

Attending lectures, carrying out exercises and the course project, and reading handouts.

Assessment methods

Oral exam (50%) with continuous assessment during the semester based on course project (50%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

Lecture notes, selected papers and book chapters distributed at each lecture.

Websites

- https://disal.epfl.ch/teaching/distributed_intelligent_systems/

Moodle Link

- https://go.epfl.ch/ENG-466

Videos

- https://mediaspace.epfl.ch/channel/ENG-466+Distributed+intelligent+systems+-+Spring+2022/29956

Prerequisite for

R&D activities in engineering

COM-502

Dynamical system theory for engineers

Thiran Patrick

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Mechanical engineering	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal

Summary

Linear and nonlinear dynamical systems are found in all fields of science and engineering. After a short review of linear system theory, the class will explain and develop the main tools for the qualitative analysis of nonlinear systems, both in discrete-time and continuous-time.

Content

- **Introduction:** Dynamics of linear and non linear systems. Definitions; Unicity of a solution; Limit Sets, Attractors.
- **Linear Systems:** Solutions; Stability of autonomous systems, Geometrical analysis, connection with frequency domain analysis.
- **Nonlinear Systems:** Solutions; Examples. Large-scale notions of stability (Lyapunov functions). Hamiltonian systems, gradient systems. Small-scale notions of stability (Linearization; stability and basin of attraction of an equilibrium point, stability of periodic solutions, Floquet Multipliers). Graphical methods for the analysis of low-dimensional systems. Introduction to structural stability, Bifurcation theory. Introduction to chaotic systems (Lyapunov exponents); time permitting: a review of tools of measure theory to compute Lyapunov exponents.
- The class is methodology-driven. It may present some limited examples of applications, but it is not application-driven.

Keywords

Dynamical Systems, Attractors, Equilibrium point, Limit Cycles, Stability, Lyapunov Functions, Bifurcations, Lyapunov exponents.

Learning Prerequisites

Required courses

- Linear algebra (MATH 111 or equivalent).
- Analysis I, II, III (MATH 101, 106, 203 or equivalent).
- Circuits & Systems II (EE 205 or equivalent) or a Systems & Signals class (MICRO 310/311 or

equivalent).

Recommended courses

• A first-year Probabilty class, such as MATH-232, MATH-231, MATH-234(b), MATH-234(c), or equivalent.
• Analysis IV (MATH 207 or equivalent)

Important concepts to start the course

• Linear Algebra (vector spaces, matrix operations, including inversion and eigendecomposition).
• Calculus (linear ordinary differential equations; Fourier, Laplace and z-Transforms).
• Basic notions of topology.
• Basic notions of probability.

Learning Outcomes

By the end of the course, the student must be able to:

• Analyze a linear or nonlinear dynamical system.
• Anticipate the asymptotic behavior of a dynamical system.
• Assess / Evaluate the stability of a dynamical system.
• Identify the type of solutions of a dynamical sytem.
• Analyze a linear or nonlinear dynamical system
• Anticipate the asymptotic behavior of a dynamical system
• Assess / Evaluate the stability of a dynamical system
• Identify the type of solutions of a dynamical system

Teaching methods

• Lectures (blackboard), 2h per week
• Exercise session, 1h per week

Expected student activities

Exercises in class and at home (paper and pencil, and Matlab)

Assessment methods

1. Mid-term 20% (conditionally on the Covid situation allowing for it to be taken at EPFL).
2. Final exam 80%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Course notes; textbooks given as reference on the moodle page of the course.

Notes/Handbook

Course notes, exercises and solutions provided on the moodle page of the course.

Moodle Link

- https://go.epfl.ch/COM-502

Prerequisite for

Classes using methods from dynamical systems.

CS-476

Embedded system design

Kluter Ties Jan Henderikus

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Labs	2 weekly
Number of positions	

Summary

Hardware-software co-design is a well known concept in embedded system design.It is also a concept required in designing FPGA-accelerators in data-centers.This course teaches how to transform algorithms in smart hardware-software solutions.

Content

High-level architectures:

- FIFO's, LIFO's, ring-buffers, and ping-pong buffers.
- FSM-D (finite state machine data-path) structures.
- Stream processing.

Acceleration methods:

- Custom instruction set extensions.
- Hardware accelerators
- Compiler optimizations.

Implementation methods:

- Hardware-software co-design.
- Timing closure.
- Virtual prototyping.
- Bare metal versus usage of an RTOS.

Learning Prerequisites

Required courses

-

Recommended courses

- Architecture-aware programming
- CS-200 Computer Architecture or CS-208 Computer Architecture I and CS-209 Computer Architecture II

- Receive and give feedback (criticism) and respond appropriately

Expected student activities

The student agrees to do this internship with professionalism

Assessment methods

- Short internships (8 weeks) : electronical evaluation at the end of the internship
- Long internships (6 months) : electronical evaluation at the end of the internship
- Master's project in the industry : see the master's project course book

Supervision

Others	Industry supervisor EPFL supervisor

Resources

Notes/Handbook

Internship guidelines available on the internship website

Websites

- https://www.epfl.ch/schools/ic/internships/
- https://www.epfl.ch/about/recruiting/recruiting/internships/

DH-415

Ethics and law of AI

Rochel Johan Robert

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Minor in digital humanities, media and society	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Withdrawal	Unauthorized
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	**100**

It is not allowed to withdraw from this subject after the registration deadline.

Summary

This master course enables students to sharpen their proficiency in tackling ethical and legal challenges linked to Artificial Intelligence (AI). Students acquire the competence to define AI and identify ethical and legal questions linked to its conception and increased use in society.

Content

AI is used as shortcut-concept to identify a number of computational systems producing intelligent behavior, i.e., complex behavior conducive to reaching goals. AI systems are increasingly used across society. They raise conceptual issues (how to define AI?), technological-ethical issues (how should AI systems be conceived?), legal issues (how to define the responsibility of an AI system? how to regulate AI?) and social-political issues (which justice questions does the deployment of AI raise?)

The following issues will be dealt with:

- What is ethics?
- What is an AI system?
- Who is responsible for the actions of an AI system?
- What are the most pressing ethical questions in the phase of conception of AI systems?
- How should we design AI system in order to overcome ethical-legal challenges?
- Should we regulate AI?
- How should we address the consequences of the wide deployment of AI systems?

Keywords

artificial intelligence, ethics, law, data, regulation, responsibility

Learning Prerequisites

Required courses

No pre-requirement

Learning Outcomes

By the end of the course, the student must be able to:

- Define the concept of AI
- Assess / Evaluate the contexts in which AI is deployed
- Systematize general principles (law and ethics)
- Analyze the different senses/conceptions/interpretations of agency, autonomy and responsibility
- Develop principles for the conception of AI system
- Distinguish legal and ethical arguments

Transversal skills

- Demonstrate the capacity for critical thinking
- Take account of the social and human dimensions of the engineering profession.
- Respect relevant legal guidelines and ethical codes for the profession.
- Use a work methodology appropriate to the task.

Teaching methods

The course will be organized as an interactive and participative course. For the weekly course: students have to read texts and to be ready for critical discussion. For the weekly exercise: students have to engage in group discussions. The course requires reading complex texts in English.

Expected student activities

Weekly reading of preparatory texts
Active participation in class, both course and exercise

Assessment methods

Students will be assessed in the following way :
- Mid-term: students will have to answer 2 questions during class (compulsory, no grading)
- Open book written exam during the term (100% of the grade)

Supervision

Office hours	No
Assistants	Yes
Others	Upon appointment with Dr Rochel

Resources

Bibliography

All resources will be made available on moodle.
To start with: *AI Ethics* (Mark Coekelbergh, MIT 2020)

Ressources en bibliothèque

- AI Ethics / Mark Coeckelbergh

Moodle Link

- https://go.epfl.ch/DH-415

CS-489

Experience design

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Project	4 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

As we move towards a design economy, the success of new products, systems and services depend increasingly on the excellence of personal experience. This course introduces students to the notion and practice of experience design following a hands-on, studio-based approach.

Content

Experience design in practice encompasses the collection, analysis and design of users experiences based on a deep understanding of the context concerned. We will examine these processes using a series of mini-workshops, to rapidly iterate on multiple design experience options. The goal is to create a meaningful, interactive, data-driven (and possibly AI-assisted) digital interface and physical prototypes for new experiences.
We explore questions at the intersection of physical and digital architecture through an experience design approach, involving: (1) a mapping of the social dynamics surrounding an experience; (2) a critical analysis of the geographical and temporal flows (experience journeys); and (3) a detailed evaluation of the experience touch points. Based on this experience diagnosis, we propose alternative designs of experience blueprints that combine physical and digital touch points which in turn will constitute the elements of future typologies.
Our particular focus will be on information intensive typologies in the contemporary city, such as museums, libraries, airports, banks, boutiques, governments, hospitals and homes. Each year, we will investigate a different typology. Digital interfaces and augmented artifacts will be considered as possible alternatives to reconfigure the senses of perception, redistribute time, and reorchestrate the configuration of social, emotional and spatial experiences.
The seminar will combine students from both IC and ENAC to work together in a real interdisciplinary process.

Keywords

User Experience (UX) Design, Design Thinking, Journey Mapping, Optioneering, Critical Prototyping, Value Proposition

Learning Prerequisites

Required courses

Bachelor in Computer Science or equivalent

Learning Outcomes

By the end of the course, the student must be able to:

- Identify issues of experience design in relation to an actual design project

- Perform rigorous analysis of the problem space and map the design opportunities
- Develop alternative design concepts for future artifacts
- Translate design concepts into meaningful experiences through iterative prototyping at appropriate scales and levels of granularity
- Create convincing arguments for the design propositions and persuasive visual and tangible evidence

Teaching methods

Workshop, Design reviews, Presentations, Group projects

Expected student activities

Group discussion, Case studies, Design Reviews, Pin-Up, Desk Crits

Assessment methods

Grading will be based upon the quality of the projects in the preliminary workshops (30%), intermediary reviews (20%) and in the final review (50%). Projects will be reviewed and assessed based on their conceptual strength, the coherence of their translation into prototypes, their narrative clarity and experiential power, and the persuasiveness of their communication, both orally and through the presented artifacts.

Supervision

Office hours	Yes
Assistants	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-489

CS-550

Formal verification

Kuncak Viktor

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

We introduce formal verification as an approach for developing highly reliable systems. Formal verification finds proofs that computer systems work under all relevant scenarios. We will learn how to use formal verification tools and explain the theory and the practice behind them.

Content

Topics may include (among others) some of the following:

- Importance of Reliable Systems. Methodology of Formal Verification. Soundness and Completeness in Modeling and Tools. Successful Tools and Flagship Case Studies
- Review of Sets, Relations, Computability, Propositional and First-Order Logic Syntax, Semantics, Sequent Calculus.
- Completeness and Semi-Decidability for First-Order Logic. Inductive Definitions and Proof Trees. Higher-Order Logic and LCF Approach.
- State Machines. Transition Formulas. Traces. Strongest Postconditions and Weakest Preconditions.
- Hoare Logic. Inductive Invariants. Well-Founded Relations and Termination Measures
- Linear Temporal Logic. System Verilog Assertions. Monitors
- SAT Solvers and Bounded Model Checking
- Model Checking using Binary Decision Diagrams
- Loop Invariants. Hoare Logic. Statically Checked Function Contracts. Relational Semantics and Fixed-Point Semantics
- Symbolic Execution. Satisfiability Modulo Theories
- Abstract Interpretation
- Set theory for verification

Learning Prerequisites

Recommended courses

CS-320 Computer language processing

Important concepts to start the course

Discrete Mathematics (e.g. Kenneth Rosen: Discrete Mathematics and Its Applications)

Learning Outcomes

By the end of the course, the student must be able to:

- Formalize specifications
- Synthesize loop invariants
- Specify software functionality
- Generalize inductive hypothesis
- Critique current software development practices

Teaching methods

Instructors will present lectures and exercises and supervise labs on student laptops.

Expected student activities

Follow the course materials, take mid-term, and complete and explain projects during the semester.

Assessment methods

The grade is based on the written mid-term, as well as code, documentation, and explanation of projects during the semester. Specific percentages will be communicated in the first class.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- **Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511576430**
- **Aaron Bradley and Zohar Manna: The Calculus of Computation - Decision Procedures with Applications to Verification, Springer 2007.**
- Michael Huth and Mark Rayan: Logic in Computer Science - Modelling and Reasoning about Systems. Cambridge University Press 2004.
- Handbook of Model Checking, https://www.springer.com/de/book/9783319105741 Springer 2018. Including Chapter Model Checking Security Protocols by David Basin.
- Tobias Nipkow, Gerwin Klein: Concrete Semantics with Isabelle/HOL. http://concrete-semantics.org/concrete-semantics.pdf
- Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: Principles of Program Analysis. ISBN 978-3-662-03811-6. Springer 1999.
- Peter B. Andrews: An Introduction to Mathematical Logic and Type Theory (To Truth Through Proof), Springer 2002.
- http://logitext.mit.edu/tutorial

Ressources en bibliothèque

- Handbook of Practical Logic and Automated Reasoning / Harrison
- Handbook of model checking / Clarke
- [chapter] Model Checking Security Protocols / Bassin
- Principles of Program Analysis / Flemming
- The Calculus of Computation / Bradley
- Logic in Computer Science / Huth
- Introduction to mathematical logic and type theory / Andrews

Notes/Handbook

https://lara.epfl.ch/w/fv

Websites

- https://lara.epfl.ch/w/fv

Moodle Link

- https://go.epfl.ch/CS-550

CS-459

Foundations of probabilistic proofs

Chiesa Alessandro

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	4 weekly
Exercises	1 weekly
Number of positions	

Summary

Probabilistic proof systems (eg PCPs and IPs) have had a tremendous impact on theoretical computer science, as well as on real-world secure systems. They underlie delegation of computation protocols and hardness of approximation. This course covers the foundations of probabilistic proof systems.

Content

Proofs are at the foundations of mathematics, and verifying the correctness of a mathematical proof is a fundamental computational task. (The P versus NP problem, which deals precisely with the complexity of proof verification, is one of the most important open problems in all of mathematics.) The complexity-theoretic study of proof verification has led to new notions of mathematical proofs, such as Interactive Proofs, Probabilistically Checkable Proofs, and others.
Probabilistic proofs are a powerful tool for proving hardness of approximation results, and are an essential building block to achieve delegation of computation (protocols that enable super fast verification of long computations, such as SNARKs). Via these applications, probabilistic proofs have had a tremendous impact on theoretical computer science and, more recently, are playing an exciting role in applied cryptography, computer security, and blockchain technology (e.g., they help secure billions of dollars in transactionds per day).
This course provides an introduction to probabilistic proofs and the beautiful mathematics underlying them.
Covered topics include arithmetization, the sumcheck protocol, zero knowledge, doubly-efficient interactive proofs, linearity testing, low-degree testing, proof composition, succinct verification, and more.
This courses assumes basic familiarity with algorithms (asymptotic notation and analysis of algorithms), complexity theory (computation models and simple complexity classes), and some algebra (finite fields and their properties).

Learning Prerequisites

Important concepts to start the course

- Basic knowledge of discrete probability.
- Basic knowledge of finite fields and their properties.
- Basic knowledge of algorithms (asymptotic notation and analysis of algorithms).
- Basic knowledge of computational complexity (Turing machines; boolean circuits; complexity classes; reductions, familiarity with the classes P and NP; probabilistic computation and the class BPP).

Teaching methods

Two weekly lectures that include definitions, theorems, and proofs. One weekly recitation to guide students through exercises. Weekly problem sets to reinforce the material.

Resources

Moodle Link

- https://go.epfl.ch/CS-459

CS-457 **Geometric computing**

Pauly Mark

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

This course will cover mathematical concepts and efficient numerical methods for geometric computing. We will explore the beauty of geometry and develop algorithms to simulate and optimize 2D and 3D geometric models with an emphasis towards computational design for digital fabrication.

Content

- Overview of modern digital fabrication technology
- Discrete geometric models for curves, surfaces, volumes
- Basics of finite element modeling
- Physics-based simulation methods
- Forward and inverse design optimization methods
- Shape Optimization

Keywords

geometry, simulation, shape optimization, digital fabrication

Learning Prerequisites

Recommended courses

CS-328 Numerical Methods for Visual Computing and ML

Important concepts to start the course

Undergraduate knowledge of linear algebra, calculus, and numerical methods; programming experience (e.g.
Python, C/C++, Java, Scala)

Learning Outcomes

By the end of the course, the student must be able to:

- Model and formalize geometric shape design & optimization problems
- Design and implement computational methods for shape processing, physics-based simulation, and numerical optimization based on discrete geometry representations
- Apply geometric abstraction principles to reduce the complexity of shape optimization problems
- Assess / Evaluate geometry processing algorithms for their suitability for specific digital fabrication technologies

Transversal skills

• Demonstrate a capacity for creativity.
• Continue to work through difficulties or initial failure to find optimal solutions.
• Use both general and domain specific IT resources and tools
• Evaluate one's own performance in the team, receive and respond appropriately to feedback.

Teaching methods

Lectures, interactive demos, exercises, practical work sessions

Expected student activities

Attend and participate in lectures, study provided reading material, solve theory exercises and implementation homeworks, design and fabricate (with support) physical models

Assessment methods

Graded homeworks, final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

• https://go.epfl.ch/CS-457

EPFL

MATH-483

Gödel and recursivity

Duparc Jacques

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Cours donné en alternance tous les deux ans

Summary

Gödel incompleteness theorems and mathematical foundations of computer science

Content

Recursivity :
Turing Machines and variants. The Church-Turing Thesis. Universal Turing Machine. Undecidable problems (the halting and the Post-Correspondance problems). Reducibility. The arithmetical hierarchy. Relations to Turing machines. Turing degrees.
Gödel's theorems:
Peano and Robinson Arithmetics. Representable functions. Incompleteness, and undecidability theorems in arithmetics.

If time permits, the completeness of the first order theory of the field of real numbers.

Keywords

Gödel, incompleteness theorems, Peano arithmetic, Robinson arithmetic, decidability, recursively enumerable, arithmetical hierarchy, Turing machine, Turing degrees, jump operator, primitive recursive functions, recursive functions, automata, pushdown automata, regular languages, context-free languages, recursive languages, halting problem, universal Turing machine, Church thesis.

Learning Prerequisites

Recommended courses

Mathematical logic (or equivalent). A good understanding of 1st order logic is required - in particular the relation between syntax and semantics.

Important concepts to start the course

1st order logic: syntax, semantics, proof theory, completeness theorem, compactness theorem, Löwenheim-Skolem theorem.

Learning Outcomes

By the end of the course, the student must be able to:

- Estimate whether a given theory, function, language is recursive or no
- Decide the class that a language belongs to (regular, context-free, recursive,...)
- Elaborate an automaton
- Design a Turing machine
- Formalize a proof in Peano arithmetic
- Sketch the incompleteness theorems
- Propose a non-standard model
- Argue why Hilbert program failed

Teaching methods

Ex cathedra lecture and exercises

Assessment methods

Written: 3 hours
Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Set Theory:
- Thomas Jech: Set theory, Springer 2006
- Kenneth Kunen: Set theory, Springer, 1983
- Jean-Louis Krivine: Theory des ensembles, 2007
- Patrick Dehornoy: Logique et théorie des ensembles; Notes de cours, FIMFA ENS: http://www.math.unicaen.fr/~dehornoy/surveys.html
- Yiannis Moschovakis: Notes on set theory, Springer 2006
- Karel Hrbacek and Thomas Jech: Introduction to Set theory, (3d edition), 1999

Recursion Theory :
- Micheal Sipser: Introduction to the Theory of Computation, Thomson Course Technology Boston, 2006
- Piergiorgio Odifreddi: Classical recursion theory, vol. 1 and 2, Springer, 1999
- Robert I. Soare: Recursively Enumerable Sets and Degres, A Study of Computable Functions and Computably Generated Sets, Springer-Verlag 1987
- Nigel Cutland: Computability, an introduction to recursive function theory, 1980
- Raymond M. Smullyan: recursion theory for methamathematics, Oxford, 1993

Proof theory :
- Wolfram Pohlers: Proof Theory, the first step into impredicativity, Springer, 2008
- A. S. Troelstra, H. Schwichtenberg, and Anne S. Troelstra: Basic proof theory, Cambridge, 2000

- S.R. Buss: Handbook of proof theory, Springer, 1998

Gödel's results :
- Raymond M. Smullyan: Gödel's incompleteness theorems, Oxford, 1992
- Peter Smith: An introduction to Gödel's theorems, Cambridge, 2008
- Torkel Franzen: Inexhaustibility, a non exhaustive treatment, AK Peteres, 2002
- Melvin Fitting: Incompleteness in the land of sets, King's College, 2007
- Torkel Franzen: Gödel's theorem: an incomplete guide to its use and abuse, AK Peters, 2005

Ressources en bibliothèque

- Théorie des ensembles / Krivine
- Inexhaustibility, a non exhaustive treatment / Franzen
- Proof Theory / Pohlers
- Notes on theory / Moschovakis
- Basic proof theory / Troelstra
- Introduction to the Theory of Computation / Sipser
- Handbook of proof theory / Buss
- Set theory / Jech
- Classical recursion theory / Odifreddi
- Recursion theory for methamathematics / Smullyan
- Set theory / Kunen
- Incompleteness in the land of sets / Fitting
- Recursively Enumerable Sets and Degres / Soare
- Gödel's theorem / Franzen
- Computability, an introduction to recursive function theory / Cutland
- Logique et théorie des ensembles / Dehornoy
- Gödel's incompleteness theorems / Smullyan
- An introduction to Gödel's theorems / Smith
- Introduction to Set theory / Hrbacek

Websites

- http://www.hec.unil.ch/logique/enseignement/recursivity

Moodle Link

- https://go.epfl.ch/MATH-483

MICRO-511

Image processing I

Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Minor in Imaging	H	Opt.
Minor in life sciences engineering	H	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Nuclear engineering	MA1	Opt.
Photonics minor	H	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly
Number of positions	

Summary

Introduction to the basic techniques of image processing. Introduction to the development of image-processing software and to prototyping using Jupyter notebooks. Application to real-world examples in industrial vision and biomedical imaging.

Content

- **Introduction**. Image processing versus image analysis. Applications. System components.
- **Characterization of continuous images**. Image classes. 2D Fourier transform. Shift-invariant systems.
- **Image acquisition**. Sampling theory. Acquisition systems. Histogram and simple statistics. Max-Lloyd quantization (K-means).
- **Characterization of discrete images and linear filtering**. z-transform. Convolution. Separability. FIR and IIR filters.
- **Morphological operators**. Binary morphology (opening, closing, etc.). Gray-level morphology.
- **Image-processing tasks**. Preprocessing. Matching and detection. Feature extraction. Segmentation.
- **Convolutional neural networks**. Basic components. Operator-based formalism. CNN in practice: denoising and segmentation.

Learning Prerequisites

Required courses

Signals and Systems I & II (or equivalent)

Important concepts to start the course

1-D signal processing: convolution, Fourier transform, z-transform

Learning Outcomes

By the end of the course, the student must be able to:

- Exploit the multidimensional Fourier transform
- Select appropriately Hilbert spaces and inner-products
- Optimize 2-D sampling to avoid aliasing
- Formalize convolution and optical systems
- Design digital filters in 2-D
- Analyze multidimensional linear shift-invariant systems
- Apply image-analysis techniques
- Construct image-processing software
- Elaborate morphological filters
- Exploit the multidimensional Fourier transform
- Select appropriately Hilbert spaces and inner-products
- Optimize 2-D sampling to avoid aliasing
- Formalize convolution and optical systems
- Design digital filters in 2-D
- Analyze multidimensional linear shift-invariant systems
- Apply image-analysis techniques
- Construct image-processing software

Transversal skills

- Use a work methodology appropriate to the task.
- Manage priorities.
- Use both general and domain specific IT resources and tools

Assessment methods

- 70% final exam
- 30% IP labs during semester

Resources

Moodle Link

- https://go.epfl.ch/MICRO-511

MICRO-512

Image processing II

Liebling Michael, Sage Daniel, Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Environmental Sciences and Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Photonics minor	E	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly
Number of positions	

Summary

Study of advanced image processing; mathematical imaging. Development of image-processing software and prototyping in Jupyter Notebooks; application to real-world examples in industrial vision and biomedical imaging.

Content

- **Directional image analysis.** Mathematical foundations. Structure tensor. Steerable filters.
- **Continuous representation of discrete data.** Splines. Interpolation. Geometric transformations. Multi-scale decomposition (pyramids and wavelets).
- **Image transforms.** Karhunen-Loève transform (KLT). Discrete cosine transform (DCT). JPEG coding. Image pyramids. Wavelet decomposition.
- **Reconstruction in the continuum.** Wiener filter. Radon transform. Fourier slice theorem. Filtered backprojection.
- **Computational imaging.** Imaging as an inverse problem. Iterative reconstruction methods. Elements of convex analysis. Regularization & sparsity constraints.

Learning Prerequisites

Required courses

Image Processing I
Recommended courses
Signals and Systems I & II, linear algebra, analysis
Important concepts to start the course
Basic image processing and related analytical tools (Fourier transform, z-tranform, etc.)

Recommended courses

Signals and Systems I & II, linear algebra, analysis

Important concepts to start the course

Basic image processing and related analytical tools (Fourier transform, z-tranform, etc.)

Learning Outcomes

- Construct interpolation models and continuous-discrete representations
- Analyze image transforms
- Design image-reconstruction algorithms
- Formalize multiresolution representations using wavelets
- Design deconvolution algorithms
- Perform image analysis and feature extraction
- Design image-processing software (plugins)
- Synthesize steerable filters
- Construct interpolation models and continuous-discrete representations
- Analyze image transforms
- Design image-reconstruction algorithms
- Formalize multiresolution representations using wavelets
- Perform image analysis and feature extraction
- Design image-processing software
- Design image reconstruction algorithms

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Manage priorities.
- Access and evaluate appropriate sources of information.
- Use both general and domain specific IT resources and tools

Assessment methods

The objectives of the course will be assessed as follows:

- 70% final exam
- 30% IP labs

Resources

Moodle Link

- https://go.epfl.ch/MICRO-512

CS-487

Industrial automation

Sommer Philipp Alexander, Tournier Jean-Charles

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Energy Science and Technology	MA2, MA4	Opt.
Mechanical engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	**36**

It is not allowed to withdraw from this subject after the registration deadline.

Remark

This course can be taken by students of all engineering sections.

Summary

This course consists of two parts: 1) architecture of automation systems, hands-on lab 2) dependable systems and handling of faults and failures in real-time systems, including fault-tolerant computing

Content

Trends like digitalization and internet of things affect the way industrial plants are designed, deployed and operated. Industrial Automation comprises the control, communication and software architecture in (real-time) automation systems: factories, energy production and distribution, vehicles and other embedded systems.

Keywords

1. Processes and plants, automation system architecture
2. Instrumentation, Programmable Logic Controllers and embedded computers
3. Industrial communication networks, field busses
4. Field device access protocols and application program interfaces
5. Human interface and supervision
6. Dependability (Reliability, Availability, Safety, ...)

Learning Prerequisites

Recommended courses

Communication networks

Learning Outcomes

By the end of the course, the student must be able to:

- Characterize the (software) architecture of a automation system
- Apply methods and trade-offs in real-time systems
- Analyze a plant
- Propose suitable automation solutions meeting the requirements

• Analyze the reliability, availability, safety of a system

Transversal skills

• Write a scientific or technical report.
• Use both general and domain specific IT resources and tools
• Communicate effectively with professionals from other disciplines.
• Keep appropriate documentation for group meetings.
• Access and evaluate appropriate sources of information.
• Plan and carry out activities in a way which makes optimal use of available time and other resources.

Teaching methods

Oral presentation aided by slides, exercises as part of the lecture, practical work (workshop at Siemens and group assignment).

Expected student activities

• Understand material presented during lectures by asking questions and/or independent (online) searches
• Attend Siemens workshop (one full day on Siemens premises in Renens)
• Work on group assignment
• Hand-in artifacts for assignment on time

Assessment methods

Assignment 25% and final oral exam 75%

Resources

Bibliography

Introduction to Industrial Automation, Stamatios Manesis & George Nikolakopoulos, CRC Press, 2018

Ressources en bibliothèque

• Introduction to Industrial Automation / Manesis

Moodle Link

• https://go.epfl.ch/CS-487

COM-402

Information security and privacy

Payer Mathias

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
SC master EPFL	MA1, MA3	Obl.
Statistics	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

This course provides an overview of information security and privacy topics. It introduces students to the knowledge and tools they will need to deal with the security/privacy challenges they are likely to encounter in today's world. The tools are illustrated with relevant applications.

Content

- Overview of cyberthreats
- Basic exploitation of vulnerabilities
- Authentication, access control, compartmentalization
- Basic applied cryptography
- Operational security practices and failures
- Machine learning and privacy
- Data anonymization and de-anonymization techniques
- Privacy enhancing technologies
- Blockchain and decentralization

Keywords

security, privacy, protection, intrusion, anonymization, cryptography

Learning Prerequisites

Required courses

COM-301 Computer security
Basic systems programming (in C/C++) or better
Basic networking knowledge
Good scripting knowledge (Python)

Learning Outcomes

By the end of the course, the student must be able to:

- Understand the most important classes of information security/privacy risks in today's "Big Data" environment
- Exercise a basic, critical set of "best practices" for handling sensitive information
- Exercise competent operational security practices in their home and professional lives
- Understand at overview level the key technical tools available for security/privacy protection
- Understand the key technical tools available for security/privacy protection
- Exercise competent operational security practices

Expected student activities

Attending lectures, solving assigned problems and "hands-on" exercises, reading and demonstrating understanding of provided materials.

Assessment methods

- continuous control : 30% of the grade
- final exam : 70% of the grade

Resources

Moodle Link

- https://go.epfl.ch/COM-402

COM-404

Information theory and coding

Telatar Emre

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Minor in Quantum Science and Engineering	H	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

The mathematical principles of communication that govern the compression and transmission of data and the design of efficient methods of doing so.

Content

1. Mathematical definition of information and the study of its properties.
2. Source coding: efficient representation of message sources.
3. Communication channels and their capacity.
4. Coding for reliable communication over noisy channels.
5. Multi-user communications: multi access and broadcast channels.
6. Lossy source coding : approximate representation of message sources.
7. Information Theory and statistics

Learning Outcomes

By the end of the course, the student must be able to:

- Formulate the fundamenal concepts of information theory such as entropy, mutual information, channel capacity
- Elaborate the principles of source coding and data transmission
- Analyze source codes and channel codes
- Apply information theoretic methods to novel settings

Teaching methods

Ex cathedra + exercises

Assessment methods

With continuous control

Resources

Websites

- http://moodle.epfl.ch/enrol/index.php?id=14593

Moodle Link

- https://go.epfl.ch/COM-404

CS-430

Intelligent agents

Faltings Boi

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	3 weekly
Number of positions	

Summary

Software agents are widely used to control physical, economic and financial processes. The course presents practical methods for implementing software agents and multi-agent systems, supported by programming exercises, and the theoretical underpinnings including computational game theory.

Content

The course contains 4 main subject areas:

1) Basic models and algorithms for individual agents:
Models and algorithms for rational, goal-oriented behavior in agents: reactive agents, reinforcement learning, exploration-exploitation tradeoff, AI planning methods.
2) Multi-agent systems:
multi-agent planning, coordination techniques for multi-agent systems, distributed algorithms for constraint satisfaction.
3) Self-interested agents:
Models and algorithms for implementing self-interested agents motivated by economic principles: elements of computational game theory, models and algorithms for automated negotiation, social choice, mechanism design, electronic auctions and marketplaces.
4) Implementing multi-agent systems:
Agent platforms, ontologies and markup languages, web services and standards for their definition and indexing.

Learning Prerequisites

Recommended courses

Intelligence Artificielle or another introductory course to AI

Learning Outcomes

By the end of the course, the student must be able to:

- Choose and implement methods for rational decision making in software agents, based on decision processes and AI planning techniques
- Choose and implement methods for efficient rational decision making in teams of multiple software agents
- Model scenarios with multiple self-interested agents in the language of game theory
- Evaluate the feasibility of achieving goals with self-interested agents using game theory

- Design, choose and implement mechanisms for self-interested agents using game theory
- Implement systems of software agents using agent platforms

Teaching methods

Ex cathedra, practical programming exercises

Expected student activities

Lectures: 3 hours
Reading: 3 hours
Assignments/programming: 4 hours

Assessment methods

Midterm and quizzes 30%, final exam 70%

Resources

Bibliography

Michael Wooldridge : An Introduction to MultiAgent Systems - Second Edition, John Wiley & Sons, 2009
Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach (2nd/3rd/4th Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009/2015.

Ressources en bibliothèque

- An Introduction to MultiAgent Systems / Wooldridge
- Artificial Intelligence: A Modern Approach / Russell

Moodle Link

- https://go.epfl.ch/CS-430

CS-486

Interaction design

Pu Pearl

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course focuses on goal-directed design and interaction design, two subjects treated in depth in the Cooper book (see reference below). To practice these two methods, we propose a design challenge, which is further divided into mini-projects evenly spaced throughout the semester.

Content

Design methods for HCI
What is HCI: its aims and goals
Design thinking
Goal-directed Design
Mental model and different types of users
Qualitative research and user interviews
User modeling: persona and empathy diagram
Scenarios, requirements and framework design
Visual design
Information Visualization design
Basic prototyping methods for HCI
Storyboarding
Context scenario
Interactive prototype
Video prototype
Human computer interaction evaluation methods
Cognitive walkthrough
Heuristic evaluation
Evaluation with users

Keywords

Interaction design, design thinking, user interviews, ideation, storyboard, context scenarios, digital mockup, user evaluation, video prototyping.

Learning Prerequisites

Required courses

Interaction personne-système

Recommended courses

Open to students enrolled in the Master and PhD programs in IC.

Important concepts to start the course

Goal-directed design, design thinking, user needs assessment, user interviews & observation, ideation, prototyping, evaluation

Learning Outcomes

By the end of the course, the student must be able to:

- Interview users and elicit their needs using the goal-directed design method
- Design and implement interfaces and interactions
- Project management : set objectives and device a plan to achieve them
- Group work skills : discuss and identify roles, and assume those roles including leadership
- Communication : writing and presentation skills

Teaching methods

Lectures, flipped classroom lectures, exercises, hands-on practice, case studies

Expected student activities

Participation in lectures, exercises, user interviews, ideation sessions, readings, design project, project presentation

Assessment methods

The assessments consist of five in-class open-book exercises, each lasting one hour. Three of these exercises will be randomly selected for grading. Additionally, there will be two mini-projects that will be graded based on group performance. Furthermore, students' individual engagement in group activities, including user interviews, ideation, prototyping, and peer evaluation, will also be evaluated to determine individual performance.
30% open-book exercises (done in class, open notes, open book) - individual performance
20% individual engagement in group activities such as user interviews - individual performance
50% project - group performance

Resources

Bibliography

About Face 3: The Essentials of Interaction Design by Alan Cooper et al. (available as e-book at NEBIS)

Ressources en bibliothèque

- About Face 3 / Cooper

Moodle Link

- https://go.epfl.ch/CS-486

Videos

- https://mediaspace.epfl.ch/channel/CS-486%2BInteraction%2BDesign/29793?&&

CS-428 Interactive theorem proving

Barrière Aurèle, Pit-Claudel Clément

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	2 weekly
Number of positions	

Summary

A hands-on introduction to interactive theorem proving, computer-checked mathematics, compiler verification, proofs as programs, dependent types, and proof automation. Come learn how to write computer-checked proofs and certified bug-free code!

Content

- Intro to the Coq proof assistant (logic, higher-order functions, tactics)
- Functional programming (inductive types and fixpoints)
- Structural induction (data structures and verified algorithms)
- Interpreter-based program semantics (intro to compiler verification)
- Inductive relations (predicates, rule induction)
- Automation and tactics I (bottom-up reasoning and logic programming)
- Operational program semantics (small-and big-step semantics)
- Program logics (hoare triples)
- Automation and tactics II (top-down reasoning)
- Type systems (Simply-typed lambda calculus)
- Dependent types and equality proofs
- Automation and tactics III (proofs by reflection)
- Real-world theorem proving (various topics)

Learning Prerequisites

Recommended courses

This course assumes no knowledge of programming language theory. The following courses may be useful, but are not required:

- CS-320 Computer language processing (to introduce the concept of interpreter)
- CS-425 Foundations of software (to introduce type systems and the lambda calculus)
- CS-550 Formal verification (for a different perspective on theorem proving)

Important concepts to start the course

- Functional programming

Learning Outcomes

- Implement purely-functional algorithms in the Gallina language
- Translate informal requirements about software into precise mathematical properties
- Plan and carry out mechanized proofs in Coq (e.g. maths, algorithms, compilers, type systems)
- Automate repetitive proof tasks by crafting simple custom decision procedures

Teaching methods

- Lectures
- Live-coding sessions

Expected student activities

- Lectures
- Programming and verification assignments
- Project (proposal, check-in, presentation, report)

Assessment methods

- Take-home programming and verification assignments: 40% of the final grade (3 or 4 labs)
- Formal verification project: 60% of the final grade (~10 weeks, in teams of 1 to 4 students)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-511

CS-491 **Introduction to IT consulting**

Regev Gil

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	6 weekly
Number of positions	

Summary

This course is an introduction to the alignment of enterprise needs with the possibilities offered by Information Technology (IT). Using a simulated business case, we explore how to define the requirements for an IT service that matches stakeholders explicit and implicit wishes.

Content

Target Audience

Engineers who want to become
- Business Analysts
- Requirements Engineers
- Project Managers
- Management and IT consultants
- Product Owners

Content

Technological and societal changes are pressuring enterprise IT departments to hire engineers with excellent technical andbusiness skills. Their roles are called business analysts, requirements engineers, or product owners. Their skills enable the bidirectional alignment of business needs and IT capabilities. With IT being the most important enabler of enterprise strategy, these roles are crucial in many organizations, large and small, private or public.

We use experiential learning beginning with concrete experience, followed by reflection and abstraction to encourage collaborative learning by doing. You will be part of a small team that needs to understand and solve a business case through fast-paced role-playing with the teaching staff. This is interspersed with lectures on the nature of organizations, business analysis and the role of enterprise IT. Several external speakers from industry illustrate what we see in class.

We will explore the following subjects:

- Problems and solutions
- Requirements elicitation
- Business process modeling
- Project management
- Change management
- Enterprise and service modeling
- The nature of organizations
- Creating a request for tender

Keywords

Appreciation, business analysis, business process, business service, contextual inquiry,
ethnography, homeostasis, interviews, IT service, motivation modeling, request for tender, requirements engineeing,

resilience, service modeling

Learning Outcomes

By the end of the course, the student must be able to:

- Elicit requirements with business stakeholders
- Analyze business stakeholder perception and motivations
- Assess / Evaluate business processes
- Define requirements for business and IT services
- Present problems and solutions to management

Transversal skills

- Demonstrate a capacity for creativity.
- Communicate effectively with professionals from other disciplines.
- Take feedback (critique) and respond in an appropriate manner.

Teaching methods

Experimental learning and teamwork.

Assessment methods

Group oral exam.

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Beyer, H. and K. Holtzblatt (1999). "Contextual design." interactions 6(1): 32-42.
Markus M.L., Keil M. (1994). If We Build It, They Will Come: Designing Information Systems that People Want to use, Sloan Management Review; Summer 1994; 35, 4; ABI/INFORM Global pg. 11
Regev, G. et al.(2013) What We Can Learn about Business Modeling from Homeostasis, Lecture Notes in Business Information Processing, 142, 1-15, 2003
Zachman, J. A. (1987). "A framework for information systems architecture." IBM Syst. J. 26 (3): 276-292.
Weinberg, G.M., The secrets of consulting, Dorset House, 1985

Ressources en bibliothèque

- A framework for information systems architecture / Zachman
- What We Can Learn about Business Modeling from Homeostasis / Regev
- Contextual design / Holtzblatt

Moodle Link

- https://go.epfl.ch/CS-491

CS-431

Introduction to natural language processing

Bosselut Antoine, Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The objective of this course is to present the main models, formalisms and algorithms necessary for the development of applications in the field of natural language information processing. The concepts introduced during the lectures will be applied during practical sessions.

Content

Several models and algorithms for automated textual data processing will be described: morpho-lexical level: n-gram and language models, spell checkers, ...; semantic level: models and formalisms for the representation of meaning, embedings, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computationnal Linguisitics; Part-of-Speech tagging

Learning Outcomes

By the end of the course, the student must be able to:

- Compose key NLP elements to develop higher level processing chains
- Assess / Evaluate NLP based systems
- Choose appropriate solutions for solving typical NLP subproblems (tokenizing, tagging, ...)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independant components

Teaching methods

Flipped classroom (reviews and supervised "hands-on" in class) ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 16%, final exam 84%

Resources

Bibliography

1. M. Rajman editor, "*Speech and Language Engineering*", EPFL Press, 2006.
2. Daniel Jurafsky and James H, Martin, "*Speech and Language Processing"*, Prentice Hall, 2008 (2nd edition)
3. Christopher D. Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing*", MIT Press, 2000
4. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008
5. Nitin Indurkhya and Fred J. Damerau editors, "*Handbook of Natural Language Processing*", CRC Press, 2010 (2nd edition)

Ressources en bibliothèque

- Handbook of Natural Language Processing / Indurkhya
- Introduction to Information Retrieval / Manning
- Foundations of Statistical Natural Language Processing / Manning
- Speech and Language Engineering / Rajman
- Speech and Language Processing / Jurafsky

Websites

- https://coling.epfl.ch/

Moodle Link

- https://go.epfl.ch/CS-431

CS-479

Learning in neural networks

Gerstner Wulfram

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Oral
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Summary

Artificial Neural Networks are inspired by Biological Neural Networks. One big difference is that optimization in Deep Learning is done with the BackProp Algorithm, whereas in biological neural networks it is not. We show what biologically plausible learning algorithms can do (and what not).

Content

- Why BackProp is biologically not plausible. Biological two-factor rules and neuromorphic hardware
- Hebbian Learning (two-factor rules) for PCA and ICA
- Two-factor rules for dictionary learning (k-means/competitive learning/winner-takes-all)
- Three-factor rules and neuromodulators (theory and neuroscience)
- Three-factor rules for reward-based learning (theory)
- Three-factor rules for TD reinforcement-learning (algorithmic formulations)
- Actor-critic networks
- Reinforcement learning in the brain
- Learning by surprise and novelty: exploration and changing environments (algorithmic)
- Surprise and novelty in the brain
- Learning representations in multi-layer networks (algorithms without backprop)
- Learning to find a goal: a bio-plausible model with place cells and rewards
- Neuromorphic hardware and in-memory computing

Keywords

- Hebbian learning and two-factor rules
- distributed local algorithms,
- Principal Component Analysis/Independent Component Analysis (PCA and ICA)
- Reinforcement Learning (RL)
- surprise and novelty
- three-factor rules
- neuromorphic hardware

Learning Prerequisites

Required courses

Linear Algebra AND Analysis.
Machine learning

Recommended courses

Signal processing

Important concepts to start the course

Optimization, Gradient Descent, Filtering, Loss function, Eigenvalues,

Learning Outcomes

By the end of the course, the student must be able to:

- Translate concepts from machine learning and signal processing into bio-plausible algorithms
- Translate neuroscience of learning into algorithms
- Explain differences between and similarities of various algorithms
- Discriminate imitations and advantages of various learning algorithms for implementation in biology or hardware

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Set objectives and design an action plan to reach those objectives.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Give feedback (critique) in an appropriate fashion.
- Manage priorities.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Ex cathedra, Exercises, and Miniproject

Expected student activities

Participation in Class, Solution of Exercises, Miniproject.

Assessment methods

Oral exam (70 percent) plus miniproject (30 percent). If more than 45 students participate, the oral exam is replaced by a written exam.

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-479

CS-526

Learning theory

Macris Nicolas

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Machine learning and data analysis are becoming increasingly central in many sciences and applications. This course concentrates on the theoretical underpinnings of machine learning.

Content

- Basics : statistical learning framework, Probably Approximately Correct (PAC) learning, learning with a finite number of classes, Vapnik-Chervonenkis (VC).
- Bias-variance tradeoff and modern double descent phenomena.
- Neural Nets : representation power of neural nets.
- Stochastic gradient descent, modern aspects: mean field approach, neural tangent kernel.
- Matrix factorization, Tensor decompositions and factorization, Jenrich's tehorem, Alternating least squares, Tucker decompositions.
- Applications: e.g. Learning mixture models, topic modeling.

Learning Prerequisites

Recommended courses

- Analysis I, II, III
- Linear Algebra
- Machine learning
- Probability
- Algorithms (CS-250)

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the framework of PAC learning
- Explain the importance basic concepts such as VC dimension, bias-variance tradeoff and double descent
- Describe basic facts about representation of functions by neural networks
- Describe recent results on specific topics e.g., matrix and tensor factorization, learning mixture models

Teaching methods

- Lectures
- Exercises

Expected student activities

- Attend lectures
- Attend exercises sessions and do the homework

Assessment methods

Final exam and graded homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Course website

EPFL

CS-433 **Machine learning**

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computational biology minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Digital Humanities	MA1, MA3	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Opt.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Neuro-X minor	H	Opt.
Neuro-X	MA1, MA3	Opt.
Quantum Science and Engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**8 weekly**
Lecture	4 weekly
Exercises	2 weekly
Project	2 weekly
Number of positions	

Summary

Machine learning methods are becoming increasingly central in many sciences and applications. In this course, fundamental principles and methods of machine learning will be introduced, analyzed and practically implemented.

Content

1. *Basic regression and classification concepts and methods: Linear models, overfitting, linear regression, Ridge regression, logistic regression, k-NN, SVMs and kernel methods*
2. *Fundamental concepts: cost-functions and optimization, cross-validation and bias-variance trade-off, curse of dimensionality.*
3. *Neural Networks: Representation power, backpropagation, activation functions, CNN, regularization, data augmentation, dropout*
4. *Unsupervised learning: k-means clustering, gaussian mixture models and the EM algorithm. Basics of self-supervised learning*
5. *Dimensionality reduction: PCA and matrix factorization, word embeddings*
6. *Advanced methods: Adversarial learning, Generative adversarial networks*

Keywords

- *Machine learning, pattern recognition, deep learning, neural networks, data mining, knowledge discovery, algorithms*

Learning Prerequisites

Required courses

• Analysis I, II, III
• Linear Algebra
• Probability and Statistics (MATH-232)
• Algorithms I (CS-250)

Recommended courses

• *Introduction to machine learning (CS-233)*
• *...or similar bachelor lecture from other sections*

Important concepts to start the course

• *Basic probability and statistics (conditional and joint distribution, independence, Bayes rule, random variables, expectation, mean, median, mode, central limit theorem)*
• *Basic linear algebra (matrix/vector multiplications, systems of linear equations, SVD)*
• *Multivariate calculus (derivative w.r.t. vector and matrix variables)*
• *Basic Programming Skills (labs will use Python)*

Learning Outcomes

By the end of the course, the student must be able to:

• Define the following basic machine learning problems: Regression, classification, clustering, dimensionality reduction, time-series
• Explain the main differences between them
• Implement algorithms for these machine learning models
• Optimize the main trade-offs such as overfitting, and computational cost vs accuracy
• Implement machine learning methods to real-world problems, and rigorously evaluate their performance using cross-validation. Experience common pitfalls and how to overcome them
• Explain and understand the fundamental theory presented for ML methods
• Conduct a real-world interdisciplinary machine learning project, in collaboration with application domain experts
• Define the following basic machine learning models: Regression, classification, clustering, dimensionality reduction, neural networks, time-series analysis

Teaching methods

• Lectures
• Lab sessions
• Course Projects

Expected student activities

Students are expected to:

- attend lectures
- attend lab sessions and work on the weekly theory and coding exercises
- work on projects using the code developed during labs, in small groups

Assessment methods

- Written final exam
- Continuous control (Course projects)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Christopher Bishop, Pattern Recognition and Machine Learning
- Kevin Murphy, Machine Learning: A Probabilistic Perspective
- Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning
- Michael Nielsen, Neural Networks and Deep Learning
- (Jerome Friedman, Robert Tibshirani, Trevor Hastie, The elements of statistical learning : data mining, inference, and prediction)

Ressources en bibliothèque

- The elements of statistical learning : data mining, inference, and prediction / Friedman
- Pattern Recognition and Machine Learning / Bishop
- Understanding Machine Learning / Shalev-Shwartz
- Machine Learning / Murphy

Références suggérées par la bibliothèque

- Neural Networks and Deep Learning / Nielsen

Notes/Handbook

https://github.com/epfml/ML_course

Websites

- https://www.epfl.ch/labs/mlo/machine-learning-cs-433/

Moodle Link

- https://go.epfl.ch/CS-433

CS-421

Machine learning for behavioral data

Käser Tanja

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Remark

Cours biennal

Summary

Computer environments such as educational games, interactive simulations, and web services provide large amounts of data, which can be analyzed and serve as a basis for adaptation. This course will cover the core methods of user modeling and personalization, with a focus on educational data.

Content

The users of computer environments such as intelligent tutoring systems, interactive games, and web services are often very heterogeneous and therefore it is important to adapt to their specific needs and preferences.
This course will cover the core methods of adaptation and personalization, with a focus on educational data. Specifically we will discuss approaches to the task of accurately modeling and predicting human behavior within a computer environment. Furthermore, we we will also discuss data mining techniques with the goal to gain insights into human behavior. We will cover the theories and methodologies underlying the current approaches and then also look into the most recent developments in the field.

1. Cycle of adaptation : representation, prediction, intervention (e.g. recommendation)
2. Data Processing and Interpretation (missing data, feature transformations, distribution fitting)
3. Performance evaluation (cross-validation, error measures, statistical significance, overfitting)
4. Representation & Prediction (probabilistic graphical models, recurrent neural networks, logistic models, clustering-classification approaches)
5. Recommendation (collaborative filtering, content-based recommendations, multi-armed bandits)
6. Stealth Assessment (seemless detection of user traits)
7. Multimodal analytics (represent & analyze data from non-traditional sources. i.e. sensors, classroom analytics, human-robot interaction)

Learning Prerequisites

Required courses

The student must have passed a course in probability and statistics and a course including a programming project

Recommended courses

- CS-433 Machine learning or
- CS-233 Introduction to machine learning

Important concepts to start the course

Probability and statistics, basic machine learning knowledge, algorithms and programming, Python

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the main machine learning approaches to personalization, describe their advantages and disavantages and explain thedifferences between them.
- Implement algorithms for these machine learning models
- Apply them to real-world data
- Assess / Evaluate their performance
- Explain and understand the fundamental theory underlying the presented machine learning models

Teaching methods

- Lectures
- Weekly lab sessions
- Course project

Expected student activities

- Attend the lectures
- Attend the lab sessions and work on the homework assignments
- Project work

Assessment methods

- Project work (50%)
- Final exam (50%)

Resources

Moodle Link

- https://go.epfl.ch/CS-421

EPFL

MGT-427

Management de projet et analyse du risque

Wieser Philippe

Cursus	Sem.	Type
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Génie civil	MA1, MA3	Obl.
Informatique	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Microtechnique	MA1, MA3	Opt.
Mineur en Management, technologie et entrepreneuriat	H	Opt.
Robotique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Langue d'enseignement	français
Crédits	4
Retrait	Non autorisé
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Projet	1 hebdo
Nombre de places	

Il n'est pas autorisé de se retirer de cette matière après le délai d'inscription.

Résumé

Ce cours a pour objectif de présenter l'approche générale du management de projet en intégrant la gestion du risque dans toutes les étapes du projet.

Contenu

Le projet est un processus consistant à réaliser un système dans le but de satisfaire le besoin de futurs utilisateurs. Il sera ainsi aborder dans ce cours la présentation des différentes phases du projet, les organisations de projet, les moyens et méthodes de développement des variantes de projets, le choix multicritère de la variante à réaliser, la planification et le suivi dans la phase de réalisation. Une approche « risque » sera abordée à chaque étape du projet par une méthodologie probabiliste.

- Les principes généraux du management de projet
- Les phases d'un projet
- L'approche stratégique et opérationnel d'un projet
- Le management des risques
- Les formes organisationnelles de projets
- Evaluation physique de variantes de projets
- Evaluation comportementale de variantes de projets par simulation numérique
- Evaluation économique de projet
- choix de variante(s), analyse multicritère
- Planification et ordonnancement de projet
- Exemples et études de cas
- Applications informatiques

Mots-clés

Management de projet, risques, évaluation économique, planification, analyse multicritère

Compétences requises

Cours prérequis obligatoires

Néant

Cours prérequis indicatifs

Notions de base de statisques

Concepts importants à maîtriser

Ouvert, curieux et capable d'aborder un domaine complexe multidisciplinaire et multiculturel.
L'étudiant devra avoir une vision transversale des processus et être capable de raisonner de manière systémique

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Diriger une équipe de projet
- Elaborer des variantes de projet
- Planifier les variantes de projet
- Sélectionner ou choisir la variante retenue
- Anticiper les risques
- Organiser
- Restituer et communiquer
- Implémenter

Compétences transversales

- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Etre responsable des impacts environnementaux de ses actions et décisions.
- Dialoguer avec des professionnels d'autres disciplines.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Recueillir des données.
- Faire une présentation orale.

Méthode d'enseignement

Ex cathedra, projet avec présentation orale

Travail attendu

Suivi des cours et étude des documents de cours distribués
Réalisation d'un projet durant le semestre. Ce projet est réalisé en groupe et présenté dans les dernières séances du cours

Méthode d'évaluation

- 50% projet durant le semestre
- 50% examen final

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Non

Autres	Disponibilité de l'enseignant par email, téléphone ou visite à son bureau

Ressources

Bibliographie

Indiquée en rapport avec chaque chapitre du cours

Polycopiés

Cours et copies des slides de présentation envoyés à chaque étudiant sous format électronique

Liens Moodle

- https://go.epfl.ch/MGT-427

Préparation pour

Travaux de semestre
Projet de Master

COM-516 **Markov chains and algorithmic applications**

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Labs	1 weekly
Number of positions	

Remark

Pas donné en 2024-25

Summary

The study of random walks finds many applications in computer science and communications. The goal of the course is to get familiar with the theory of random walks, and to get an overview of some applications of this theory to problems of interest in communications, computer and network science.

Content

Part 1: Markov chains (~6 weeks):
- basic properties: irreducibility, periodicity, recurrence/transience, stationary and limiting distributions,
- ergodic theorem: coupling method
- detailed balance
- convergence rate to the equilibrium, spectral gap, mixing times
- cutoff phenomenon

Part 2: Sampling (~6 weeks)
- classical methods, importance and rejection sampling
- Markov Chain Monte Carlo methods, Metropolis-Hastings algorithm, Glauber dynamics, Gibbs sampling
- applications: function minimization, coloring problem, satisfiability problems, Ising models
- coupling from the past and exact simulation

Keywords

random walks, stationarity, ergodic, convergence, spectral gap, mixing time, sampling, Markov chain Monte Carlo, coupling from the past

Learning Prerequisites

Required courses

Basic probability course
Basic linear algebra and calculus courses

Recommended courses

Stochastic Models for Communications (COM-300)

Important concepts to start the course

Good knowledge of probability and analysis.
Having been exposed to the theory of Markov chains.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze the behaviour of a random walk
- Assess / Evaluate the performance of an algorithm on a graph
- Implement efficiently various sampling methods

Teaching methods

ex-cathedra course

Expected student activities

active participation to exercise sessions and implementation of a sampling algorithm

Assessment methods

midterm (20%), mini-project (20%), final exam (60%)

Resources

Bibliography

Various references will be given to the students during the course, according to the topics discussed in class.

Ressources en bibliothèque

- Probability and random processes / Grimmett

Notes/Handbook

Lecture notes will be provided

Websites

- https://moodle.epfl.ch/course/view.php?id=15016

Moodle Link

- https://go.epfl.ch/COM-516

Prerequisite for

This course is not so to speak a prerequisite for other courses, but could complement well the course COM-512 on Networks out of control, as well as other courses in statistics.

EPFL

COM-405 **Mobile networks**

Al Hassanieh Haitham

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course provides a detailed description of the organization and operating principles of mobile and wireless communication networks, as well as the use of wireless signals for sensing and imaging.

Content

- Introduction to wireless networks
- Wireless PHY Layer Techniques
- MAC (Medium Access Control) Layer Protocols
- Wi-Fi & Bluetooth
- Cellular networks (3G, 4G, 5G).
- Internet of Things (IoT) Networks and Technologies.
- Multi-Hop Networks, Mesh Networks, and Sensor Networks
- Routing in Wireless Networks
- Network Coding
- Cross Layer Networking
- Wireless Localization
- Wireless Sensing
- Wireless Imaging

Keywords

Communication networks, protocols, wireless, IoT

Learning Prerequisites

Required courses

CS-202 Computer systems or COM-208 Computer Networks
COM-302 Principles of Digital Communications

Recommended courses

COM-430 Modern Digital Communications: A Hands-on Approach

Important concepts to start the course

Operating principles of communication protocols and layer organization.

Learning Outcomes

By the end of the course, the student must be able to:

- Synthesize the way a mobile network operates
- Interpret the behavior of such networks
- Propose evolutions to existing protocols
- Identify weaknesses and bottlenecks

Teaching methods

Lectures
Weekly Readings
Exercise sessions
Homework Problems
Labs

Expected student activities

Class Participation, Quizzes, Homework, Labs, Exercise Sessions

Assessment methods

Homeworks
Quizzez
Labs
Final exam

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Handouts, recommended books (see course URL)

Ressources en bibliothèque

-

Moodle Link

- https://go.epfl.ch/COM-405

EPFL

COM-430 Modern digital communications: a hands-on approach

Chiurtu Nicolae

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course complements the theoretical knowledge learned in PDC with more advanced topics such as OFDM, MIMO, fading channels, and GPS positioning. This knowledge is put into practice with hands-on exercises based on Matlab or Python (at choice) and on a software-defined radio platform.

Content

1. Software radio : key concepts.
2. Matlab/Python implementation of the signal processing chain to the level of detail taught in Principles of Digital Communications (PDC: COM-302).
3. Channel modeling, estimation, equalization.
4. Implementation of a basic wireless communication system using a software-defined radio testbed.
5. Fading and diversity.
6. OFDM and MIMO : theory and implementation.
7. CDMA in the context of a GPS system.
8. Decoding of a GPS signal and positioning.

Keywords

Wireless, OFDM, Diversity, Coding, GPS, CDMA, MMSE, Rayleigh fading, software-defined radio, channel estimation.

Learning Prerequisites

Required courses

COM-302 Principles of Digital Communications (PDC) or equivalent.

Important concepts to start the course

Solid understanding of linear algebra and probability as well as real and complex analysis.

Learning Outcomes

By the end of the course, the student must be able to:

- Design and implement an advanced digital communication system (data rate, spectral bandwidth, energy requirements, error probability, implementation complexity).
- Model the physical properties of wired and wireless communication channels.
- Implement various parts of a "physical-layer" digital communication system.
- Understand what software-defined radio is all about.

Teaching methods

Ex cathedra lectures and small projects.

Expected student activities

Follow lectures; guided as well as independent work on projects.

Assessment methods

Written and practical midterm and final exam during the semester.
40% midterm exam, 60% final exam.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Notes/Handbook

Lecture notes

Moodle Link

- https://go.epfl.ch/COM-430

CS-552

Modern natural language processing

Bosselut Antoine

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computer and Communication Sciences		Opt.
Computer science minor	E	Obl.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

Natural language processing is ubiquitous in modern intelligent technologies, serving as a foundation for language translators, virtual assistants, search engines, and many more. In this course, students will learn algorithmic tools for tackling problems in modern NLP.

Content

This course includes lectures, exercises, a midterm exam, and a project. In lectures, we will cover the foundations of modern methods for natural language processing, such as word embeddings, recurrent neural networks, transformers, pretraining, and how they can be applied to important tasks in the field, such as machine translation and text classification. We will also cover issues with these state-of-the-art approaches (such as robustness, interpretability, sensitivity), identify their failure modes in different NLP applications, and discuss analysis and mitigation techniques for these issues.
In assignments, students will be evaluated on their ability to implement methods learned in class on closed-form problems developed by the course staff. In their project, students will be expected to apply techniques learned in lecture to a problem of the course staff's choosing. They will formulate the problem as an NLP task, propose a suitable evaluation to measure their progress, develop a model to solve the task, and provide analysis of the strengths and weaknesses of their method.
This course is of interest to MS / PhD student interested in modern methods and issues in natural language processing, both from a research and applied perspective.

Learning Prerequisites

Recommended courses

- CS-233a or CS-233b Introduction to machine learning
- CS-433 Machine learning

Important concepts to start the course

- Python programming
- Probability and Statistics
- Linear Algebra
- Machine Learning concepts

Learning Outcomes

By the end of the course, the student must be able to:

- Define basic problems and tasks in natural language processing (e.g., machine translation, summarization, text classification, language generation, sequence labeling, informaiton extraction, question answering)
- Implement common modern approaches for tackling NLP problems and tasks (embeddings, recurrent neural models, attentive neural models) and how to train them
- Understand failure modes of these models and learning algorithms (e.g., robustness, interpretability/explainability, bias, evaluation)
- Review academic research papers and understand their contributions, strengths, and weaknesses according to the principles learned in lecture
- Complete a project that applies these algorithms to a real-world NLP problem, where they will define a task, evaluation, model implementation, and analyze the shortcomings of their approach

Teaching methods

- Lectures
- Lab sessions
- Midterm Examination
- Course project

Expected student activities

- Attend lectures and participate in class
- Complete homework assignments
- Pass midterm exam
- Complete a project (set by the course supervisor) : complete a project proposal outlining topic and evaluation plan; submit two project milestones; submit final project report; present project findings to committee of course instructor and TAs.

Assessment methods

- Midterm examination (30%)
- Project (70%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-552

EE-452

Network machine learning

Frossard Pascal, Thanou Dorina

Cursus	Sem.	Type
Computational and Quantitative Biology		Opt.
Computational biology minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

Fundamentals, methods, algorithms and applications of network machine learning and graph neural networks

Content

Context

In the last decade, our information society has mutated into a data society, where the volume of worldwide data grows increasingly fast. An increasing amount of this data is structured on networks of different forms. How to make sense of such tremendous volume of data? Developing effective techniques to extract meaningful information from large-scale and high-dimensional network datasets has become essential for the success of business, government and science.

Objective

The goal of this course is to provide a broad introduction to effective methods algorithms in data science, network analysis and network machine learning. A major effort will be given to show that existing data analysis techniques can be defined and enhanced on graphs. Graphs can encode complex structures like cerebral connection, stock exchange, and social network. Strong mathematical tools have been developed based on statistics, or linear and non-linear graph spectral harmonic analysis to advance the standard data analysis algorithms. At the same time, modern machine learning tools such as neural networks have been adapted to process data defined on network structures. The objective of the class is to develop fundamentals and review algorithms that permit to develop modern network data analysis methods. The main topics of the course are networks, network data analysis, unsupervised and supervised learning on graphs and networks, graph generative models, sparse representation, multi-resolution analysis, graph neural networks.

Structure

The course is organized into two parts: lectures (2 hours) and lab assignements and projects (1 hour). The essential objective of the exercises and lab assignements is to apply the theory on real-world cases. The objective of the projects is to study practical network machine learning cases, and develop effective solutions based on tools studied in the class.

Evaluation

Evaluation will be conducted on a continuous basis: homeworks and coding assignments.

Keywords

graph representation learning, machine learning, network science

Learning Prerequisites

Required courses

Fundamentals of Machine Learning, or equivalent
Signal Processing, or equivalent
Introduction to Statistics, or equivalent
Python programming

Learning Outcomes

By the end of the course, the student must be able to:

- Apply modern machine learning techniques to network data
- Analyze network properties, network data distributions, and properties of the most common network machine learning algorithms
- Propose solutions for network data analysis problems

Transversal skills

- Use a work methodology appropriate to the task.
- Give feedback (critique) in an appropriate fashion.
- Communicate effectively, being understood, including across different languages and cultures.

Resources

Moodle Link

- https://go.epfl.ch/EE-452

COM-512

Networks out of control

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Data science minor	E	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - Pas donné en 2024-25

Summary

The goal of this class is to acquire mathematical tools and engineering insight about networks whose structure is random, as well as learning and control techniques applicable to such network data.

Content

- Random graph models: Erdös-Renyi, random regular, geometric, percolation, small worlds, stochastic block model
- Learning graphs from data: centrality metrics, embeddings, Hawkes processes, network alignment
- Control of processes on graphs: epidemics, navigation

Keywords

Random graphs, network data, machine learning, graph processes.

Learning Prerequisites

Required courses

Stochastic models in communication (COM-300), or equivalent.

Important concepts to start the course

Basic probability and stastistics; Markov chains; basic combinatorics.

Teaching methods

Ex cathedra lectures, exercises, mini-project

Expected student activities

Attending lectures, bi-weekly homeworks, mini-project incl. student presentation at the end of semester, final exam.

Assessment methods

1. Homeworks 10%
2. Mini-project 40%
3. Final exam 50%.

Resources

Bibliography

- A. D. Barbour, L. Holst and S. Janson, Poisson Approximation, Oxford Science Publications, 1992.
- B. Bollobas, Random Graphs (2nd edition), Cambridge University Press, 2001.
- R. Durrett, Random Graph Dynamics, Cambridge University Press, 2006 (electronic version).
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010 (electronic version).
- G. Grimmett, Percolation (2nd edition), Springer, 1999.
- S. Janson, T. Luczak, A. Rucinski, Random Graphs, Wiley, 2000.
- R. Meester and R. Roy, Continuum Percolation, Cambridge University Press, 1996.

Ressources en bibliothèque

- Random Graphs / Bollobas
- Random Graphs / Janson
- Continuum Percolation / Meester
- Percolation / Grimmett
- Networks, Crowds and Markets / Easley
- Poisson Approximation / Barbour
- Random Graph Dynamics / Durrett

Notes/Handbook

Class notes will be available on the course website.

Moodle Link

- https://go.epfl.ch/COM-512

MATH-489

Number theory II.c - Cryptography

Jetchev Dimitar

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The goal of the course is to introduce basic notions from public key cryptography (PKC) as well as basic number-theoretic methods and algorithms for cryptanalysis of protocols and schemes based on PKC.

Content

Basic notions and algorithms from public key cryptography such as RSA, ElGamal, key exchange protocols, zero knowledge proofs. Main topics may include, but are not limited to

- modular and finite field arithmetic
- primality testing
- polynomial and integer factorization algorithms
- index calculus and discrete logarithm-based schemes
- elliptic curve arithemtic and cryptography
- basic notions from lattice-based cryptography: lattice-basis reduction algorithms, learning-with-errors, applications to homomorphic encryption

Keywords

public key cryptography, key exchange, digital signatures, zero knowledge proofs, RSA, ElGamal, integer factorization, index calculus, elliptic curve cryptography, lattice-based cryptography

Learning Prerequisites

Recommended courses

Some knowledge of abstract algebra (groups, rings and fields) is desirable, but not mandatory. Knowledge of basic algorithms is a plus.

Assessment methods

Homework assignments: Weekly problem sets focusing on number-theoretic and complexity-theoretic aspects. These will be complemented by programming exercises in SAGE which is a Python-based computer algebra system. No prior experience with SAGE or Python is required. A subset of the homework will be handed in and graded, counting for 40% of the final grade.
The written **final exam** counts for 60% of the final grade. There will be a mock midterm exam not to be part of the final grade. The final exam will test theoretical understanding as well as understanding of the algorithms and protocols. The exam will include no SAGE programming exercises. If needed, algorithms could be presented with pseudo-code. The exact final exam format will be adapted to the epidemiological situation and resulting guidelines.

Resources

Moodle Link

- https://go.epfl.ch/MATH-489

CS-439 # Optimization for machine learning

Flammarion Nicolas

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Quantum Science and Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.
Statistics	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	1 weekly
Number of positions	

Summary

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.

Content

This course teaches an overview of modern optimization methods, for applications in machine learning and data science. In particular, scalability of algorithms to large datasets will be discussed in theory and in implementation.
Fundamental Contents:

- Convexity, Gradient Methods, Proximal algorithms, Stochastic and Online Variants of mentioned methods, Coordinate Descent Methods, Subgradient Methods, Non-Convex Optimization, Frank-Wolfe, Accelerated Methods, Primal-Dual context and certificates, Lagrange and Fenchel Duality, Second-Order Methods, Quasi-Newton Methods, Gradient-Free and Zero-Order Optimization.

Advanced Contents:

- Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods, Alternating minimization for matrix and tensor factorizations
- Parallel and Distributed Optimization Algorithms, Synchronous and Asynchronous Communication
- Lower Bounds

On the practical side, a graded **group project** allows to explore and investigate the real-world performance aspects of the algorithms and variants discussed in the course.

Keywords

Optimization, Machine learning

Learning Prerequisites

Recommended courses

- CS-433 Machine Learning

Important concepts to start the course

- Previous coursework in calculus, linear algebra, and probability is required.
- Familiarity with optimization and/or machine learning is useful.

Learning Outcomes

By the end of the course, the student must be able to:

- Assess / Evaluate the most important algorithms, function classes, and algorithm convergence guarantees
- Compose existing theoretical analysis with new aspects and algorithm variants.
- Formulate scalable and accurate implementations of the most important optimization algorithms for machine learning applications
- Characterize trade-offs between time, data and accuracy, for machine learning methods

Transversal skills

- Use both general and domain specific IT resources and tools
- Summarize an article or a technical report.

Teaching methods

- Lectures
- Exercises with Theory and Implementation Assignments

Expected student activities

Students are expected to:

- Attend the lectures and exercises
- Give a short scientific presentation about a research paper
- Read / watch the pertinent material
- Engage during the class, and discuss with other colleagues

Assessment methods

- Continuous control (course project)
- Final Exam

Resources

Websites

- https://github.com/epfml/OptML_course

Moodle Link

- https://go.epfl.ch/CS-439

Videos

- https://www.youtube.com/playlist?list=PL4O4bXkI-fAeYrsBqTUYn2xMjJAqlFQzX

COM-507

Optional research project in communication Systems

Profs divers *

Cursus	Sem.	Type
Communication systems minor	E, H	Opt.
SC master EPFL	MA1, MA2, MA3, MA4	Opt.

Language of teaching	English
Credits	8
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**2 weekly**
Project	2 weekly
Number of positions	

Remark

for students doing a minor in Communication systems : Registration upon approval of the section. Only for 2nd year Master students. Supervision by an IC authorized professor

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Supervisor and subject to be chosen among the themes proposed on the web site :

Projects by laboratory

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Transversal skills

- Write a scientific or technical report.
- Write a literature review which assesses the state of the art.

Teaching methods

Individual and independant work, under the guidance of a professor or an assistant.

Expected student activities

Written report due within the alloted time.
Information on the format and the content of the report is provided by the project supervisor.

Assessment methods

Autumn : The written report must be returned to the laboratory no later than the Friday of the second week after the

end of the classes.
Spring : The written report must be returned to the laboratory no later thant **the Friday of the first week** after the end of the classes.
The oral presentation is organized by the laboratory.

Resources

Websites

- https://www.epfl.ch/schools/ic/education/master/master-project/

Moodle Link

- https://go.epfl.ch/COM-507

EPFL

CS-522 **Principles of computer systems**

Argyraki Katerina, Candea George

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	During the semester
Workload	240h
Weeks	14
Hours	**4 weekly**
Lecture	4 weekly
Number of positions	

Summary

This advanced graduate course teaches the key design principles underlying successful computer and communication systems, and shows how to solve real problems with ideas, techniques, and algorithms from operating systems, networks, databases, programming languages, and computer architecture.

Content

A modern computer system spans many layers: applications, libraries, operating systems, networks, and hardware devices. Building a good system entails making the right trade-offs (e.g., between performance, durability, and correctness) and understanding emergent behaviors. Great system designers make these trade-offs in a principled fashion, whereas average ones make them by trial-and-error. In this course we develop a principled framework for computer system design, covering the following topics:

- Modules and interfaces
- Names
- Layers
- Indirection and virtualization
- Redundancy and fault tolerance
- Client/server architectures
- Decentralized architectures
- Transactional building blocks

Learning Prerequisites

Required courses

The course is intellectually challenging and fast-paced, and it requires a solid background in operating systems, databases, networking, programming languages, and computer architecture. The basic courses on these topics teach how the elemental parts of modern systems work, and this course picks up where the basic courses leave off. To do well, a student must master the material taught in the following courses:

- CS-200 Computer architecture
- CS-214 Software construction
- CS-311 The Software enterprise - from ideas to product
- CS-300 Data-intensive systems

Recommended courses

The following EPFL courses cover material that significantly helps students taking this course, however they are not strictly required:

- CS-320: Computer language processing
- CS-470: Advanced computer architecture
- CS-422: Database systems
- COM-407: TCP/IP networking

Learning Outcomes

By the end of the course, the student must be able to:

- Design computer and communication systems that work well
- Make rational design trade-offs (e.g., performance vs. correctness, latency vs. availability)
- Anticipate emergent system behaviors (e.g., failure cascades, security vulnerabilities)
- Integrate multiple techniques, ideas, and algorithms from different fields of computing/communication into a working system

Teaching methods

- A combination of online and in-class lectures
- Interactive design sessions
- Reading assignments
- Homework assignments

Expected student activities

- Attend lectures and design sessions
- Complete the reading and writing assignments
- Participate actively in the course (physically and online)

Assessment methods

- 50% OPs
- 50% final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

See course website for the latest information and an up-to-date bibliography.

Ressources en bibliothèque

- Principles of computer system design : an introduction / Saltzer

Websites

- https://pocs.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-522

COM-416

Research project in communication systems II

Profs divers *

Cursus	Sem.	Type
SC master EPFL	MA1, MA2, MA3, MA4	Obl.

Language of teaching	English
Credits	12
Session	Winter, Summer
Semester	Fall
Exam	During the semester
Workload	360h
Weeks	14
Hours	**2 weekly**
Project	2 weekly
Number of positions	

Summary

Individual research during the semester under the guidance of a professor or an assistant.

Content

Subject to be chosen among the themes proposed on the web site :
Projects by laboratory

Learning Outcomes

By the end of the course, the student must be able to:

- Organize a project
- Assess / Evaluate one's progress through the course of the project
- Present a project

Transversal skills

- Write a literature review which assesses the state of the art.
- Write a scientific or technical report.

Expected student activities

Written report due within the allotted time.
Information on the format and the content of the report is provied by the project supervisor.

Assessment methods

Autumn : The written report must be returned to the laboratory no later than **the Friday of the second week** after the end of classes.
Spring : The written report must be returned to the laboratory no later thant **the Friday of the first week** after the end of classes.
The oral presentation is organized by the laboratory.

Resources

Websites

- https://www.epfl.ch/schools/ic/education/master/semester-project-msc/

Moodle Link

- https://go.epfl.ch/COM-416

CS-412 **Software security**

Payer Mathias

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	During the semester
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	2 weekly
Labs	1 weekly
Number of positions	

Summary

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with security pitfalls.

Content

This course focuses on software security fundamentals, secure coding guidelines and principles, and advanced software security concepts. Students will learn to assess and understand threats, learn how to design and implement secure software systems, and get hands-on experience with common security pitfalls.
Software running on current systems is exploited by attackers despite many deployed defence mechanisms and best practices for developing new software. In this course students will learn about current security threats, attack vectors, and defence mechanisms on current systems. The students will work with real world problems and technical challenges of security mechanisms (both in the design and implementation of programming languages, compilers, and runtime systems).

- Secure software lifecycle: design, implementation, testing, and deployment
- Basic software security principles
- Reverse engineering : understanding code
- Security policies: Memory and Type safety
- Software bugs and undefined behavior
- Attack vectors: from flaw to compromise
- Runtime defense: mitigations
- Software testing: fuzzing and sanitization
- Focus topic: Web security
- Focus topic: Mobile security

Keywords

Software security, mitigation, software testing, sanitization, fuzzing

Learning Prerequisites

Required courses

- COM-402 Information security and privacy (or an equivalent security course)
- A systems programming course (with focus on C/C++)
- An operating systems course

Important concepts to start the course

Basic computer literacy like system administration, build systems, C/C++ programming skills, debugging, and development skills. Understanding of virtual machines and operating systems.

Learning Outcomes

By the end of the course, the student must be able to:

- Explain the top 20 most common weaknesses in software security and understand how such problems can be avoided in software.
- Identify common security threats, risks, and attack vectors for software systems.
- Assess / Evaluate current security best practices and defense mechanisms for current software systems. Become aware of limitations of existing defense mechanisms and how to avoid them.
- Identify security problems in source code and binaries, assess the associated risks, and reason about their severity and exploitability.
- Assess / Evaluate the security of given source code or applications.

Transversal skills

- Identify the different roles that are involved in well-functioning teams and assume different roles, including leadership roles.
- Keep appropriate documentation for group meetings.
- Summarize an article or a technical report.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Make an oral presentation.

Teaching methods

The lectures are denser early in the semester, then tapering off before the end. They are backed up by PDF files of all the lecture material, as well as a few textbook recommendations.
The exercises sessions start slowly early in the semester but pick up and occupy all time towards the end. Homework exercises consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.
The labs focus on practical software security aspects and during the course the students will be assessed through their

completion of several challenging "hands on" labs.

Expected student activities

Students are encouraged to attend lectures and exercise sessions. In addition to normal studying of the lecture and practice of the exercises, the reading assignment consists of analyzing a few suggested scientific papers on a large selection of topics; the presentation assignment consists of holding a 15-minute presentation on the selected topic; and the writing assignment of documenting what was learned in a term paper due at the end of the semester.

Assessment methods

The grade will continuously be evaluated through a combination of pracitcal assignments in the form of several labs and theoretical quizzes throughout the semester. The labs will account for 50%, the quizzes and tests to 50%.
The exact dates of the labs/quizzes will be communicated at the beginning of the class.

Resources

Notes/Handbook

Software Security: Principles, Policies, and Protection (SS3P, by Mathias Payer)
https://nebelwelt.net/SS3P/

Moodle Link

- https://go.epfl.ch/CS-412

MATH-486 **Statistical mechanics and Gibbs measures**

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Pas donné en 2024-25. Cours donné en alternance une année sur deux.

Summary

This course provides a rigorous introduction to the ideas, methods and results of classical statistical mechanics, with an emphasis on presenting the central tools for the probabilistic description of infinite lattice systems.

Content

The goals of this course are to present

- the probabilistic description of large systems with interacting components,
- the mathematical description of phase transitions occuring in certain discrete models (Curie-Weiss, Ising model, long-range models, etc.)
- the general theory of infinite-volume Gibbs measures (the so-called Dobrushin-Lanford-Ruelle approach)

If times permits, and depending on the interest of the participants, we consider the peculiar properties of certain models with an underlying continuous symmetry (Gaussian free field, Mermin-Wagner Theorem for O(n) models).
This course is companion to the course ``lattice models'', where discrete models
are also considered, but with an emphasis on different aspects.

The lectures will be largely based on the book *Statistical mechanics of lattice systems; a concrete mathematical introduction*, by S. Friedli and Y. Velenik (Cambridge University Press, 2017)

Keywords

statistical mechanics, phase transitions, Gibbs measures, entropy, Ising model, Gaussian Free Field

Learning Prerequisites

Required courses

- Analyse 1et 2
- Théorie de la Mesure
- Probabilités

Assessment methods

Examen oral.

Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Statistical mechanics of lattice systems; a concrete mathematical introduction, by S. Friedli and Y. Velenik (Cambridge University Press, 2017)
Gibbs Measures and Phase Transitions, by H.-O. Georgii (De Gruyter Studies in Mathematics Vol. 9. Berlin: de Gruyter 1988)

Ressources en bibliothèque

- Statistical mechanics of lattice systems / Friedli & Velenik
- Gibbs Measures and Phase Transitions / Georgii
- (electronic version)

Websites

- http://www.unige.ch/math/folks/velenik/smbook/

Moodle Link

- https://go.epfl.ch/MATH-486

PHYS-512

Statistical physics of computation

Erba Vittorio

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Ing.-phys	MA1, MA3	Opt.
Physicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The students understand tools from the statistical physics of disordered systems, and apply them to study computational and statistical problems in graph theory, discrete optimisation, inference and machine learning.

Content

Interest in the methods and concepts of statistical physics is rapidly growing in fields as diverse as theoretical computer science, probability theory, machine learning, discrete mathematics, optimization, signal processing and others. Large part of the related work has relied on the use of message-passing algorithms and their connection to the statistical physics of glasses and spin glasses.

This course covers this active interdisciplinary research landscape. Specifically, we will review the statistical physics approach to problems ranging from graph theory (e.g. community detection) to discrete optimization and constraint satisfaction (e.g. satisfiability or coloring) and to inference and machine learning problems (learning in neural networks, clustering of data and of networks, compressed sensing or sparse linear regression, low-rank matrix factorization).

We will expose theoretical methods of analysis (replica, cavity, ...) algorithms (message passing, spectral methods, etc), discuss concrete applications, highlight rigorous justifications as well as present the connection to the physics of glassy and disordered systems.

This is an advanced theoretical course that is designed for students with background in mathematics, electrical engineering, computer science or physics. This course exposes advanced theoretical concepts and methods, with exercises in the analytical methods and usage of the related algorithms.

Learning Prerequisites

Important concepts to start the course

For physics students Statistical physics I and II (or equivalent) is required.
This lecture is accessible to students in mathematics, electrical engineering, computer science without any previous training in statistical physics. Those students are expected to have strong interest in theory, probabilistic approaches to analysis of algoriths, high-dimensional statistics or probabilistic signal processing.

Learning Outcomes

By the end of the course, the student must be able to:

- Analyze theoretically a range of problems in computer science and learning.
- Derive algorithms for a range of computational problems using technics stemming from statistical physics.

Teaching methods

2h of lecture + 2h of excercise

Assessment methods

Final written exam counting for 50% and graded homework during the semester counting for the other 50%.

Resources

Bibliography

Information, Physics and Computation (Oxford Graduate Texts), 2009, M. Mézard, A. Montanari
Statistical Physics of inference: Thresholds and algorithms, Advances in Physics 65, 5 2016, L. Zdeborova & F. Krzakala, available at https://arxiv.org/abs/1511.02476

Ressources en bibliothèque

- Information, Physics and Computation / Mézard

Références suggérées par la bibliothèque

- Statistical Physics of inference: Thresholds and algorithms / Zdeborova & Krzakala

Notes/Handbook

Policopié "Statistical Physics methods in Optimization & Machine Learning" by L. Zdeborova & F. Krzakala, available athttps://sphinxteam.github.io/EPFLDoctoralLecture2021/Notes.pdf

Moodle Link

- https://go.epfl.ch/PHYS-512

EPFL

COM-500 **Statistical signal and data processing through applications**

Ridolfi Andrea

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Obl.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Exercises	2 weekly
Number of positions	

Summary

Building up on the basic concepts of sampling, filtering and Fourier transforms, we address stochastic modeling, spectral analysis, estimation and prediction, classification, and adaptive filtering, with an application oriented approach and hands-on numerical exercises.

Content

1. Fundamentals of Statistical Signal and Data Processing: Signals and systems from the deterministic and the stochastic point of view; Processing and analysing signals and systems with a mathematical computing language.
2. Models, Methods, and Algorithms: Parametric and non-parametric signal models (wide sense stationary, Gaussian, Markovian, auto-regressive and white noise signals); Linear prediction and estimation (orthogonality principle and Wiener filter); Maximum likehood estimation and Bayesian a priori; Maximum a posteriori estimation.
3. Statistical Signal and Data Processing Tools for Spread Spectrum Wireless Transmission: Coding and decoding of information using position of pulses (annihilating filter approach); Spectrum estimation (periodogram, line spectrum methods, smooth spectrum methods, harmonic signals).
4. Statistical Signal and Data Processing Tools for the Analysis of Neurobiological Recordings: Poisson process for neurobiological spikes; Characterization of multi-state neurons (Markovian models and maximum likelihood estimation); Classifying firing rates of neuron (Mixture models and the EM algorithm); Hidden Markov models; Spike sorting and Principal Component Analysis.
5. Statistical Signal and Data Processing Tools for Echo Cancellation: Adaptive filtering (least mean squares and recursive least squares); Adaptive echo cancellation and denoising.

Keywords

Statistical tools, spectral analysis, prediction, estimation, annihilating filter, mixture models, principal component analysis, stochastic processes, hidden Markov models, adaptive filtering, mathematical computing language (Matlab, Python, or similar).

Learning Prerequisites

Required courses

Stochastic Models in Communications (COM-300), Signal Processing for Communications (COM-303) / Signal Processing (COM-202).

Important concepts to start the course

Calculus, Algebra, Fourier Transform, Z Transform, Probability, Linear Systems, Filters.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose appropriate statistical tools to solve signal processing problems;
- Analyze real data using a mathematical computing language;
- Interpret spectral content of signals;
- Develop appropriate models for observed signals;
- Assess / Evaluate advantages and limitations of different statistical tools for a given signal processing problem;
- Implement numerical methods for processing signals.

Transversal skills

- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking
- Access and evaluate appropriate sources of information.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Ex cathedra with exercises and numerical examples.

Expected student activities

Attendance at lectures, completing exercises, testing presented methods with a mathematical computing language (Matlab, Python, or similar).

Assessment methods

- 20% midterm
- 20% mini project
- 60% Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Background texts

- P. Prandoni, *Signal Processing for Communications*, EPFL Press;
- P. Bremaud, *An Introduction to Probabilistic Modeling*, Springer-Verlag, 1988;
- A.V. Oppenheim, R.W. Schafer, *Discrete Time Signal Processing*, Prentice Hall, 1989;
- B. Porat, *A Course in Digital Signal Processing*, John Wiley & Sons,1997;
- C.T. Chen, *Digital Signal Processing*, Oxford University Press;
- D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability,* Athena Scientific, 2002 (excellent book on probability).

More advanced texts

- L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*, Springer-Verlag, 1988;

- A.N. Shiryaev, *Probability*, Springer-Verlag, New York, 2nd edition, 1996;
- S.M. Ross, *Introduction to Probability Models*, Third edition, 1985;
- P. Bremaud, Markov Chains, Springer-Verlag, 1999;
- P. Bremaud, Mathematical Principles of Signal Processing, Springer-Verlag, 2002;
- S.M. Ross, *Stochastic Processes*, John Wiley, 1983;
- B. Porat, *Digital Processing of Random Signals*, Prentice Hall,1994;
- P.M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, 1993;
- P. Stoïca and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, 1997.

Ressources en bibliothèque

- Probability / Shiryaev
- Stochastics Processes / Ross
- Discrete Time Signal Processing / Oppenheim
- Introduction to Spectral Analysis / Stoïca
- Digital Processing of Random Signals / Porat
- Introduction to Probability / Bertsekas
- Introduction to Hilbert Spaces with Applications / Debnath
- Signal Processins for Communications / Prandoni
- An Introduction to Probabilistic Modeling / Bremaud
- A Course in Digital Signal Processing / Porat
- Optimal and Adaptive Signal Processing / Clarkson
- Digital Signal Processing / Chen
- Introduction to Probability Models / Ross

Notes/Handbook

- Slides handouts;
- Collection of exercises.

Moodle Link

- https://go.epfl.ch/COM-500

MATH-413

Statistics for data science

Chandak Rajita Ramesh, Limnios Myrto

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Managmt, tech et entr.	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 weekly
Number of positions	

Summary

Statistics lies at the foundation of data science, providing a unifying theoretical and methodological backbone for the diverse tasks enountered in this emerging field. This course rigorously develops the key notions and methods of statistics, with an emphasis on concepts rather than techniques.

Content

Keywords

Data science, inference, likelihood, regression, regularisation, statistics.

Learning Prerequisites

Required courses

Real analysis, linear algebra, probability.

Recommended courses

A first course in statistics.

Important concepts to start the course

Students taking the course will need a solid grasp of notions from analysis (limits, sequences, series, continuity, differential/integral calculus) and linear algebra (linear subspaces, bases, dimension, eigendecompositions, etc). Though the course will cover a rapid review of probability, a first encounter with the subject is necessary (random variables, distributions/densities, independence, conditional probability). Familiarity with introductory level notions of statistics would be highly beneficial but not necessary.

Learning Outcomes

By the end of the course, the student must be able to:

- Derive properties of fundamental statistical procedures
- Estimate model parameters from empirical observations
- Test hypotheses related to the structural characteristics of a model
- Construct confidence bounds for model parameters and predictions
- Contrast competing models in terms of fit and parsimony

Teaching methods

Slides and whiteboard.

Assessment methods

Final exam and a midterm counting for 15%.
Dans le cas de l'art. 3 al. 5 du Règlement de section, l'enseignant décide de la forme de l'examen qu'il communique aux étudiants concernés.

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

Davison, A.C. (2003). Statistical Models, Cambridge.
Panaretos, V.M. (2016). Statistics for Mathematicians. Birkhäuser.
Wasserman, L. (2004). All of Statistics. Springer.
Friedman, J., Hastie, T. and Tibshirani, R. (2010). Elements of Statistical Learning. Springer

Ressources en bibliothèque

- Elements of Statistical Learning
- Statistical Models
- Statistics for Mathematicians
- All of Statistics

Moodle Link

- https://go.epfl.ch/MATH-413

EPFL

COM-506 **Student seminar: security protocols and applications**

Vaudenay Serge

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	3
Session	Summer
Semester	Spring
Exam	During the semester
Workload	90h
Weeks	14
Hours	**2 weekly**
Lecture	2 weekly
Number of positions	

Summary

This seminar introduces the participants to the current trends, problems, and methods in the area of communication security.

Content

We will look at today's most popular security protocols and new kinds of protocols, techniques, and problems that will play an emerging role in the future. Also, the seminar will cover methods to model and analyze such security protocols. This course will be held as a seminar, in which the students actively participate. The talks will be assigned in the first meeting to teams of students, and each team will have to give a 45 minutes talk, react to other students' questions, and write a 3-4 pages summary of their talk.

Keywords

network security, security protocols, cryptography

Learning Prerequisites

Required courses

- Computer security (COM-301)
- Cryptography and security (COM-401)

Learning Outcomes

By the end of the course, the student must be able to:

- Synthesize some existing work on a security protocol
- Analyze a security protocol
- Present a lecture

Transversal skills

- Make an oral presentation.
- Summarize an article or a technical report.

Expected student activities

- prepare a lecture (presentation and a 4-page report)
- present the lecture
- attend to others' lectures and grade them

Assessment methods

- lecture and attendance to others' lectures

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Websites

- https://lasec.epfl.ch/teaching.php

Moodle Link

- https://go.epfl.ch/COM-506

Videos

- https://mediaspace.epfl.ch/channel/COM-506+Student+Seminar+on+Security+Protocols+and+Applications

CS-448

Sublinear algorithms for big data analysis

Kapralov Michael

Cursus	Sem.	Type
Computer and Communication Sciences		Obl.
Computer science minor	E	Obl.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal, donné les années impaires

Summary

In this course we will define rigorous mathematical models for computing on large datasets, cover main algorithmic techniques that have been developed for sublinear (e.g. faster than linear time) data processing. We will also discuss limitations inherent to computing with constrained resources.

Content

The tentative list of topics is:

Streaming: given a large dataset as a stream, how can we approximate its basic properties using a very small memory footprint? Examples that we will cover include statistical problems such as estimating the number of distinct elements in a stream of data items, finding heavy hitters, frequency moments, as well as graphs problems such as approximating shortest path distances, maximum matchings etc.;
Sketching: what can we learn about the input from a few carefully designed measurements (i.e. a `sketch') of the input, or just a few samples of the input? We will cover several results in sparse recovery and property testing that answer this question for a range of fundamental problems;
Sublinear runtime: which problems admit solutions that run faster than it takes to read the entire input? We will cover sublinear time algorithms for graph processing problems, nearest neighbor search and sparse recovery (including Sparse FFT);
Communication: how can we design algorithms for modern distributed computation models (e.g. MapReduce) that have low communication requirements? We will discuss graph sketching, a recently developed approach for designing low communication algorithms for processing dynamically changing graphs, as well as other techniques.

Keywords

streaming, sketching, sparse recovery, sublinear algorithms

Learning Prerequisites

Required courses

Bachelor courses on algorithms, complexity theory, and discrete mathematics

Important concepts to start the course

Discrete probability; mathematical maturity

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class
- Analyze space/time/communication complexity of randomized algorithms
- Prove space/time/communication lower bounds for variations of problems discussed in class
- Choose an appropriate algorithmic tool for big data problem at hand

Teaching methods

Ex cathedra, homeworks, final

Assessment methods

Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://go.epfl.ch/CS-448

EPFL

CS-473

System programming for Systems-on-chip

Kluter Ties Jan Henderikus

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Microtechnics	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

To efficiently program embedded systems an understanding of their architectures is required. After following this course students will be able to take an existing SoC, understand its architecture, and efficiently program it.

Content

Hardware elements found in embedded systems:

- Flash, tightly couples memories, SDRAM, DDR.
- IO-interfaces and protocols (RS232, I2C, I2S, SPI).
- Bus architectures.

Architecture imposed restrictions:

- Memory map and memory holes.
- Cached and non-cached regions.
- Interrupt latencies, bus latencies, task-switch latencies.

Software techniques:

- BIOS/firmware
- DMA and computation/data-transfer overlaps
- Hot-spot detection and hardware/software profiling

Learning Prerequisites

Recommended courses

CS-200 Computer architecture or CS-208 Computer architecture I; CS-209 Computer architecture II

Important concepts to start the course

- C/C++ programming skills
- Basic Verilog knowledge

Learning Outcomes

- Analyze and understand the architecture of embedded systems (SoC's)
- Write a firmware that initializes an embedded system and efficiently implement the required functionality
- Explain the different latencies present in an embedded system and how these latencies influence the execution time on the firmware
- Profile an embedded system and pin-point the hardware related and software induced hot-spots
- understand the different types of memories present in an embedded systems and how to use them
- Program the different types of I/O devices present in an embedded system and know how their protocol works

Teaching methods

Ex cathedra with practical exercises (in groups of 2 students)

Expected student activities

- Reports of practical exercises
- Written exam

Assessment methods

- Lab reports : 50%
- Final written exam : 50%

Supervision

Office hours	No
Assistants	Yes
Others	Electronic forum and Moodle

Resources

Moodle Link

- https://go.epfl.ch/CS-473

CS-460

Systems for data management and data science

Ailamaki Anastasia, Kermarrec Anne-Marie

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computational science and engineering minor	E	Opt.
Computer and Communication Sciences		Opt.
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	8
Session	Summer
Semester	Spring
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

This course is intended for students who want to understand modern large-scale data analysis systems and database systems. The course covers fundamental principles for understanding and building systems for managing and analyzing large amounts of data. It covers a wide range of topics and technologi

Content

Topics include large-scale data systems design and implementation, and specifically :

- Distributed data management systems
- Data management : locality, accesses, partitioning, replication
- Modern storage hierarchies
- Query optimization, database tuning
- Transaction management
- Data structures : File systems, Key-value stores, DBMS
- Consistency models
- Large-scale data analytics infrastructures
- Parallel Processing
- Data stream and graph processing

Learning Prerequisites

Recommended courses

- CS-107 Introduction to programming
- CS-214 Software construction
- CS-300 Data-Intensive Systems
- CS-202 Computer systems
- CS-452 Foundations of software

Important concepts to start the course

- Algorithms and data structures.
- Scala and/or Java programming languages will be used throughout the course. Programming experience in one of these languages is strongly recommanded.
- Basic knowledge or computer networking and distributed systems.

Learning Outcomes

By the end of the course, the student must be able to:

- Understand how to design big data analytics systems using state-of-the-art infrastructures for horizontal scaling, e.g., Spark
- Implement algorithms and data structures for streaming data analytics
- Decide between different storage models based on the offered optimizations enabled by each modl and the expected query workload
- Compare concurrency control algorithms, and algorithms for distributed data management
- Configure systems parameters, data layouts, and application designs for database systems
- Develop data-parallel analytics programs that make us of modern clusters and cloud offerings to scale up to very large workloads
- Analyze the trade-offs between various approaches to large-scale data management and analytics, depending on efficiency, scalability, and latency needs

Teaching methods

Lectures, project, homework, exercises and practical work

Expected student activities

- Attend lectures and participate in class
- Complete a project as per the guidelines posted by the teaching team

Assessment methods

- Project
- Midterm (as needed)
- Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

J. Hellerstein & M. Stonebraker, Readings in Database Systems, 4th Edition, 2005
R. Ramakrishnan & J. Gehrke: "Database Management Systems", McGraw-Hill, 3rd Edition, 2002.
A. Rajaraman & J. Ullman: "Mining of Massive Datasets", Cambridge Univ. Press, 2011.

Ressources en bibliothèque

- Mining of Massive Datasets / Rajaraman
- Database Management Systems / Ramakrishnan
- Readings in Database Systems / Hellerstein

Moodle Link

- https://go.epfl.ch/CS-460

COM-407 **TCP/IP networking**

Nikolopoulos Pavlos

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Obl.
Electrical and Electronical Engineering	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language of teaching	English
Credits	8
Session	Winter
Semester	Fall
Exam	Written
Workload	240h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Labs	2 weekly
Number of positions	

Summary

In the lectures you will learn and understand the main ideas that underlie and the way communication networks are built and run. In the labs you will exercise practical configurations.

Content

-- The internet architecture.
-- Layer 2 networking; switching/bridging.
-- The Internet protocol versions 4 and 6.
-- The transport layer, TCP, UDP, sockets, QUIC.
-- Routing algorithms: Link state routing, OSPF, Distance Vector routing. Interdomain routing, BGP.
-- Congestion control principles. The fairness of TCP. Application to the Internet (TCP Reno, Cubic, DCTCP, BBR).
-- Tunnels and hybrid architectures.
-- A few things about internet security.
-- Application layer protocols.

Keywords

TCP/IP
Computer Networks

Learning Prerequisites

Required courses

A first programming course (Python)

Recommended courses

An undergraduate course on Computer Networks

Learning Outcomes

By the end of the course, the student must be able to:

- Run and configure networks
- Understand the main ideas that underlie the Internet
- Write simple communicating programs
- Use communication primitives for internet and industrial applications.

Transversal skills

- Access and evaluate appropriate sources of information.
- Continue to work through difficulties or initial failure to find optimal solutions.

Teaching methods

Lectures.
Online quizzes.
Labs on student's computer.

Expected student activities

Participate in lectures
Participate in online quizzes
Make lab assignments (in the rule, every other week)

Assessment methods

Theory grade = final exam
Practice grade = average of labs
Final grade = mean of theory grade (50%) and practice grade (50%).
The research exercice may add a bonus of at most 0.5 points in 1-6 scale to the practice grade.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

"Computer Networking: A Top-Down Approach (6th or later Edition)". James F. Kurose and Keith W. Ross. 2012. Pearson.

"Computer Networking : Principles, Protocols and Practice". O. Bonaventure, open source textbook, http://inl.info.ucl.ac.be/CNP3

Ressources en bibliothèque

-
- Computer Networking / Kurose

Notes/Handbook

Slides are on moodle

Websites

- http://moodle.epfl.ch/course/view.php?id=523

Moodle Link

- https://go.epfl.ch/COM-407

EPFL

CS-510 **Topics in software security**

Payer Mathias

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	3
Session	Winter
Semester	Fall
Exam	During the semester
Workload	90h
Weeks	14
Hours	**2 weekly**
Lecture	1 weekly
Exercises	1 weekly
Number of positions	

Summary

Memory corruption and type safety flaws dominate the threat landscape. We will approach current research from three dimensions: sanitization (finding flaws through runtime monitors); fuzzing (testing software automatically); and mitigation (protecting software at runtime).

Content

Unsafe languages like C/C++ are widely used for their great promise of performance. Unfortunately, these languages are prone to a large set of different types of memory and type errors that allow the exploitation of several attack vectors such as code reuse, privilege escalation, or information leaks.
On a high level memory and type safety (and type safety) would solve all these problems. Safe languages can (somewhat) cheaply enforce these properties.
Unfortunately, these guarantees come at a high cost if retrofitted onto existing languages.
When working with unsafe languages, three fundamental approaches exist to protect against software flaws: formal verification (proving the absence of bugs), software testing (finding bugs), and mitigation (protecting against the exploitation of bugs). In this seminar, we will primarily focus on the latter two approaches. Formal verification, while giving strong guarantees, struggles to scale to large software.
This seminar explores three areas: the understanding of attack vectors,approaches to software testing, and mitigation strategies. First you need to understand what kind of software flaws exist in low level software and how those flaws can be exploited.

Learning Prerequisites

Required courses

A security course like COM-301
An operating/systems course like CS-323

Recommended courses

COM-402 Information security and privacy
CS-412 Software security

Learning Outcomes

By the end of the course, the student must be able to:

- Investigate select advanced concepts in software security
- Promote their programming and systems skills in core security topics
- Assess / Evaluate the contributions of a software security research paper

- Investigate software security research papers
- Present a research paper and lead the resulting discussion

Teaching methods

In this seminar course, students will read, prepare, and present recent research papers in the filed of software security. The papers will be discussed in class. The presenter will organize the discussion among their peers and prepare a set of discussion points.

Expected student activities

The students are expected to

- Prepare and hold the presentation of their assigned research paper
- Summarize the paper along with the class discussion after their presentation
- Participate in the presentations and discussions of the other students

Assessment methods

- Presentation : 40%
- Summary/review : 50%
- Class participation : 10%

Resources

Websites

- https://go.epfl.ch/cs510

Moodle Link

- https://go.epfl.ch/CS-510

CS-455

Topics in theoretical computer science

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - pas donné en 2024-25

Summary

The students gain an in-depth knowledge of several current and emerging areas of theoretical computer science. The course familiarizes them with advanced techniques, and develops an understanding of fundamental questions that underlie some of the key problems of modern computer science.

Content

Examples of topics that will be covered include:

• Laplacians, random walks, graph sparsification: It is possible to compress graphs while approximately preserving their spectral properties (in particular, properties of random walks)? We will cover the main results from the recent influential line of work on spectral sparsification that provides such compression schemes.

• Laplacian system solvers: given a linear system Ax=b, how quickly can we find x? We will cover nearly linear time algorithms for solving Ax=b when A is a symmetric diagonally dominant matrix (a common scenario in practice) that crucially rely on spectral graph sparsification.

• Spectral clustering: given a graph, can we find a partition of the graph into k vertex disjoint parts such that few edges cross from one part to another? This is the fundamental graph clustering problem that arises in many applications. We will cover several results on spectral graph partitioning, where one first embeds vertices of the graph into Euclidean space using the bottom few eigenvectors of the graph Laplacian, and then employs Euclidean clustering primitives to find the partition.

• Local clustering with random walks: Given a very large graph and a seed node in it, can we find a small cut that separates the seed node from the rest of the graph, without reading the entire graph? We will cover local clustering algorithms, which identify such cuts in time roughly proportional to the number of vertices on the small side of the cut, by carefully analyzing distributions of random walks in the graph.

Keywords

spectral graph theory, sparsification, clustering, random walks

Learning Prerequisites

Required courses

Bachelor courses on algorithms and discrete mathematics, mathematical maturity.

Learning Outcomes

By the end of the course, the student must be able to:

- Design efficient algorithms for variations of problems discussed in class;
- Analyze approximation quality of spectral graph algorithms;

Teaching methods

Ex cathedra, homeworks, reading

Expected student activities

Attendance at lectures, completing exercises, reading written material

Assessment methods

- Continuous control

Supervision

Office hours	Yes
Assistants	Yes
Others	Electronique forum : Yes

Resources

Bibliography

There is no textbook for the course. Notes will be posted on the course website.

Ressources en bibliothèque

- Randomized Algorithms / Motwani

Moodle Link

- https://go.epfl.ch/CS-455

EPFL

CS-444 **Virtual reality**

Boulic Ronan

Cursus	Sem.	Type
Computer science minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Opt.
Minor in digital humanities, media and society	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly
Number of positions	

Summary

The goal of VR is to embed the users in a potentially complex virtual environment while ensuring that they are able to react as if this environment were real. The course provides a human perception-action background and describes the key programming techniques for achieving efficient VR applications

Content

The first lectures focus more on the technical means (hw & sw) for acheiving the hands-on coding sessions:

- Visual display
- Interaction devices and sensors
- Software environment (UNITY3D, programming in C#)

The proportion of more theoretical VR and Neuroscience background increases over the semester:

- Key Human perception abilities, cybersickness, immersion, presence and flow
- Basic 3D interaction techniques: Magic vs Naturalism
- The perception of action
- Haptic interaction
- What makes a virtual human looking alive ?
- VR, cognitive science and true experimental design

The group project focuses on CODING a 3D VR game. So we recommend to rather attend DH-414 "Game design & prototyping" in case of lack of coding background as this latter does not require coding skills.

Spring 2024-25 will be the last instance of CS-444 in the study plan.

Keywords

3D interaction, display, sensors, immersion, presence, embodiment

Learning Prerequisites

Required courses

Mastering an Object-Oriented programming language

Recommended courses

(CS-341) Computer graphics

Important concepts to start the course

1) Object Oriented programming lies at the core of the project development in C# with Unity3D. Some programming experience with this approach is compulsory as all students will be assessed on the individual coding of some features of the project.
2) from Computer Graphics:
- perspective transformations
- representation and manipulation of 3D orientation
- 3D modelling hierarchy
- matrix algebra: translation, orientation, composition

Learning Outcomes

By the end of the course, the student must be able to:

- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersions, presence and flow
- Give an example of applications of VR in different industrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions
- Describe how the human perception-action system is exploited in VR
- Apply the concepts of immersion, presence and flow
- Give an example of applications of VR in different inustrial sectors
- Choose a method of immersion suited for a given 3D interaction context
- Explain the possible causes of cybersickness in a given VR system configuration
- Design a VR system involving 3D interactions
- Give an example of applications of VR in different industrial sectors

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Ex cathedra + Hands-on sessions on VR devices in the first half of the semester,
A mini-project in groups of 2-3 persons will have to integrate various components of 3D real-time interaction (in C# within Unity3D). The group will submit their project proposal to the course responsible TAs who will assess whether it meets the key specifications and is original enough. The proposal will include the use of some VR devices that the IIG research group will lend during the mini-project period. The project development will have to be conducted with git.

Expected student activities

exploit citation analysis tools to evaluate a scientific paper
combine 3D interaction components to produce an original 3D experience
experiment the hands-on practical work in the lab
synthesize the knowledge acquired in course and hands-on in the theoretical oral and the project oral

Assessment methods

Scientific paper study : summary of contributions and citation analysis (around week6 of the semester)
Theoretical oral exam (last week of the semester)
Project assessment through code repository, report and oral exam around the end of the semester

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

- Course notes will be updated and made available after each course, with links to key sites and on-line documents
- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2017. 3D User Interfaces: Theory and Practice. Second edition, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O'Reilly 2015

Ressources en bibliothèque

- 3D User Interfaces / Bowman
- Learning Virtual Reality / Parisi
- The VR book / Jerald

Notes/Handbook

pdf of slides are made visible after the ex-cathedra courses

Websites

- http://www.thevrbook.net/
- http://gitlab.epfl.ch

Moodle Link

- https://go.epfl.ch/CS-444

CS-503 **Visual intelligence : machines and minds**

Zamir Amir

Cursus	Sem.	Type
Civil & Environmental Engineering		Obl.
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Minor in Imaging	E	Opt.
Neuro-X minor	E	Opt.
Neuro-X	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language of teaching	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

The course will discuss classic material as well as recent advances in computer vision and machine learning relevant to processing visual data -- with a primary focus on embodied intelligence and vision for active agents.

Content

Visual perception is the capability of inferring the properties of the external world merely from the light reflected off the objects therein. This is done beautifully well by simple (e.g., mosquitoes) or complex (e.g., humans) biological organisms. They can see and *understand* the complex environment around them and *act* accordingly -- all done in an efficient and astonishingly robust way. Computer vision is the discipline of replicating this capability for machines. The progress in computer vision has brought about successful applications, such as face detection/recognition or handwriting recognition. However, a large gap to sophisticated perceptual capabilities, such as those exhibited by animals, remains.

The goal of this course is to discuss what is possible in computer vision today and what is *not*. We will overview the basic concepts in computer vision and recent advances in machine learning relevant to processing visual data and active perception. For inspiration around the missing capabilities and how to approach them, we will turn to visual perception in biological organisms.

The course includes lectures and projects. There will be a heavy emphasis on the *projects* and *hands-on experience*. The course project will be around designing, implementing, and testing a solution to a (preferably open) problem pertinent to visual perception. The students are encouraged to work in groups, self-propose a project that excites them, and go for ambitious yet feasible projects. The course staff will provide support throughout the semester with the projects. In the lectures, the students will learn about the principles of computer vision, the current limits, and the visual perception in humans and animals, which will help them with formulating their course projects. In particular, the lectures will discuss the following:

1. A recap of basic computer vision concepts: classification, detection, grouping, image transformations, optical flow, 3D from X, etc., and recent successful neural network architectures, such as Transformers.
2. Psychology/physiology of the visual system.
3. Perception-action loop: active perception and embodied vision.

The course interests masters/PhD students interested in research in computer vision, machine learning, and perceptual robotics, as well as senior undergraduate students interested in understanding state-of-the-art computer vision.

Keywords

Computer vision, Machine learning, Embodied intelligence, Robotics, Cognition, Neural networks, AI.

Learning Prerequisites

Required courses

• Machine Learning (CS-433) or Introduction to Machine Learning (CS-233) or equivalent course on the basics of machine learning.

• Deep Learning (EE-559) or Artificial Neural Networks (CS-456) or equivalent course on the basics of deep learning.

Recommended courses

• Computer vision (CS-442) or equivalent undergraduate/masters course on the basics of computer.

Important concepts to start the course

• Deep learning and machine learning.
• Python programming.
• Basics of probability and statistics.

Learning Outcomes

By the end of the course, the student must be able to:

• Define the basic concepts in computer vision, such as detection, segmentation, 3D from X, covered in the lectures.
• Explain the range of theories in psychology around visual perception, covered in the lectures.
• Design and implement computer vision/machine learning algorithms to address problems with real-world complexity.
• Design and implement proper evaluation pipelines for computer vision/machine learning algorithms to assess their performance in the real-world.
• Assess the limits and performance pitfalls of a given computer vision/machine learning algorithm, especially when facing real-world complexity

Transversal skills

• Write a scientific or technical report.
• Make an oral presentation.
• Assess progress against the plan, and adapt the plan as appropriate.
• Demonstrate the capacity for critical thinking

Teaching methods

Lectures. Lab sessions. Project Tutoring. Course Project.

Expected student activities

• In regard to the lectured material, the students are expected to study the provided reading material, actively participate in the class, engage in the discussions, and answer homework questions.

• In regard to the course project, the students are expected to formulate and implement an in-depth project, demonstrate continuous progress throughout the semester, and provide a final written report and presentation.

Assessment methods

• Project (70%) [distributed over the project proposal, milestone reports, final report and presentation]

- Homeworks (30%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Vision Science: Photons to Phenomenology, Steven Palmer, 1999.
- The Ecological Approach to Visual Perception, Jame Gibson, 1979.
- Computer Vision: Algorithms and Applications, Richard Szeliski, 2020.
- Animal Eyes, Michael Land and Dan-Eric Nilsson, 2012.

Ressources en bibliothèque

- Animal Eyes / Land
- An immense world / Yong (added)
- Vision Science / Palmer
- Ecological Approach to Visual Perception / Gibson
- Computer Vision / Szeliski

Notes/Handbook

The reference reading of different lectures will be from different books (the main ones listed above) and occasionally from papers. Resources will be provided in class. Full-text books are not mandatory.

Moodle Link

- https://go.epfl.ch/CS-503