Plan d’études

SYSTEMES DE COMMUNICATION

2 0 2 1 - 2 0 2 2

arrêté par la direction de l'EPFL le 26 mai 2021

Directeur de section	**Prof. S. Vaudenay**
Conseillers d'études :	
Année propédeutique	**--**
1ère année cycle bachelor	**--**
2ème année cycle bachelor	**Prof. C. Troncoso**
1ère année cycle master	**Prof. R. Guerraoui**
2ème année cycle master	**vacat**
Projet de master	**Prof. P. Thiran**
Responsable passerelle HES	**Mme E. Hazboun**
Délégué à la mobilité	**M. J.-L. Benz**
Coordinatrice des stages d’ingénieur	**Mme E. Hazboun**
Adjointe de la section	**Mme E. Hazboun**
Secrétaire Bachelor	**Mme M. Emery**
Secrétaire 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.

2021-2022 **SYSTÈMES DE COMMUNICATION** **Cycle Propédeutique**

Cursus commun IN-SC

Code	Matières	Type de branches	Enseignants sous réserve de modification	Sections	Semestres BA1 c	BA1 e	BA1 p	BA2 c	BA2 e	BA2 p	Coeff.	Période des épreuves	Type examen
	Bloc 1 :										**38**		
CS-101	Advanced information, computation, communication I	Spécifique	Aberer	SC	**4**	**2**					7	H	écrit
COM-102	Advanced information, computation, communication II	Spécifique	Gastpar	SC				**4**	**2**		7	E	écrit
MATH-111(	Algèbre linéaire (en français) ou	Polytechnique	Zuleta Estrugo	MA	**4**	**2**					6	H	écrit
MATH-111(	Algèbre linéaire (en anglais)		Maddocks	MA									
MATH-111(	Algèbre linéaire (classe inversée)		Urech	MA									
MATH-101(	Analyse I (en français) ou	Polytechnique	Chizat	MA	**4**	**2**					6	H	écrit
MATH-101(	Analyse I (en allemand) ou		Kressner	MA									
MATH-101(	Analyse I (en anglais)		Svaldi	MA									
MATH-101(	Analyse I (classe inversée)		Deparis	MA									
MATH-106(	Analyse II (en français) ou	Polytechnique	Lachowska	MA				**4**	**2**		6	E	écrit
MATH-106(	Analyse II (en anglais)		Marcus	MA									
PHYS-101(h	Physique générale : mécanique (en français) ou	Polytechnique	Brantut	PH	**3**	**3**					6	H	écrit
PHYS-101(e	Physique générale : mécanique (en anglais)		Manley	PH									
	Bloc 2 :										**22**		
CS-173	Digital system design	Spécifique	Kluter	IN				**4**		**2**	6	sem P	
HUM-1nn	Enjeux mondiaux	Polytechnique	Divers enseignants	CDH				**2**			2	sem P	
CS-107	Introduction à la programmation	Polytechnique	Sam	IN	**2**	**3**					5	sem A	
CS-108	Pratique de la programmation orientée-objet	Spécifique	Schinz	IN				**2**	**2**	**6**	9	sem P	
	Totaux :				**17**	**12**	**0**	**16**	**6**	**8**	**60**		
	Totaux par semaine :					**29**			**30**				

2021-2022 **SYSTÈMES DE COMMUNICATION** **Cycle Bachelor**

Code	Matières	Enseignants sous réserve de modification	Sections	Semestres												Crédits		Période des épreuves	Type examen
				BA3			BA4			BA5			BA6						
				c	e	p	c	e	p	c	e	p	c	e	p	2e	3e		
	Bloc A															**11**			
COM-208	Computer networks	Argyraki	SC	**2**	**2**											5		H	écrit
MATH-232	Probabilities and statistics	Berthier/Abbé	MA				**4**	**2**								6		E	écrit
	Bloc B															**17**			
CS-250	Algorithms	Svensson	IN	**4**	**2**											6		H	écrit
CS-208	Computer architecture I	Stojilovic	IN	**2**		**2**										4		E	écrit
CS-207	Programmation orientée système	Chappelier	IN				**1**	**2**								3		sem P	
CS-251	Theory of computation	Göös	IN				**2**	**2**								4		E	écrit
	Bloc C															**16**			
MATH-203(c)	Analyse III	Tione/Widmayer	MA	**2**	**2**											4		H	écrit
MATH-207(d)	Analyse IV	vacat	MA				**2**	**2**								4		E	écrit
EE-205	Signals and systems (for EL&IC)	Shkel	SC				**2**	**2**								4		E	écrit
PHYS-114	General physics : electromagnetism	Shchutska	PH	**2**	**2**											4		H	écrit
	Bloc D																**18**		
COM-300	Modèles stochastiques pour les communications	Thiran	SC							**4**	**2**						6	H	écrit
COM-302	Principles of digital communications	Telatar	SC										**4**	**2**			6	E	écrit
COM-303	Signal processing for communications	Prandoni	SC										**4**	**2**			6	E	écrit
	Bloc E																**7**		
MATH-310	Algebra	Lachowska	MA							**2**	**1**						3	H	écrit
COM-301	Computer security	Troncoso	IN							**2**	**1**	**1**					4	sem A	
	Groupe "projet"																**8**		
COM-307	Projet en Systèmes de Communication I	Divers enseignants								←		2			→		8	sem A ou P	
	Groupe options			←						**35**					→	**12**	**23**		
CS-308	Calcul quantique	Macris	SC										*3*	*1*			4	E	écrit
CH-160b	Chimie générale	Terrettaz	CGC							*2*	*1*						3	H	écrit
CS-209	Computer architecture II	Ienne	IN				*2*		*2*							4		E	écrit
CS-320	Computer language processing	Kuncak	IN							*2*	*2*	*2*					6	sem A	
EE-200	Électromagnétisme I : lignes et ondes	Fleury	EL							*2*	*1*						3	H	écrit
EE-201	Électromagnétisme II : calcul des champs	Fleury	EL										*2*	*1*			3	E	écrit
EE-202b	Electronique I	Zysman	SC	*2*	*1*											4		sem A	
EE-203b	Électronique II	Zysman	SC							*2*	*2*						4	sem A	
EE-381	Electronique III	Zysman	SC										*2*	*1*			3	sem P	
CS-210	Functional programming	Kuncak/Odersky	IN	*2*	*2*											5		sem A	
CS-330	Intelligence artificielle	Faltings	IN										*2*	*2*			4	sem P	
CS-213	Interaction personne système	Dillenbourg	IN				*2*		*2*							5		E	écrit
COM-308	Internet analytics	Grossglauser	SC										*2*	*1*	*2*		5	sem P	
BIO-109	Introduction aux sciences du vivant (pour IC)	Zufferey R.	SV										*4*	*2*			6	E	écrit
CS-341	Introduction to computer graphics (pas donné en 21-22)	Pauly	IN										*2*	*1*	*2*		6	E	écrit
CS-322	Introduction to database systems	Ailamaki/Koch	IN										*2*	*1*	*1*		4	E	écrit
CS-233a	Introduction to machine learning	Salzmann	IN	*2*	*2*											4		H	écrit
CS-233b	Introduction to machine learning	Fua	IN				*2*	*2*								4		E	écrit
CS-307	Introduction to multiprocessor architecture	Falsafi	IN							*2*		*1*					4	sem A	
CS-323	Introduction to operating systems	Kashyap/Payer	IN							*2*	*1*	*2*					5	H	écrit
CS-358	Making Intelligent Things	Koch	IN						*6*						*6*	6	6	sem P	sans retrait
CS-328	Numerical methods for visual computing and ML	Jakob	IN	*2*	*1*											4		H	écrit
CS-206	Parallelism and concurrency	Kashyap/Odersky	IN				*1*	*1*	*2*							4		sem P	
CS-309	Projet de Systems-on-Chip	Beuchat	IN												*3*		3	sem P	
CS-212	Projet programmation système	Chappelier/Bugnion	IN						*2*							2		sem P	
CS-306	Software development project	Candea	IN												*4*		4	sem P	
CS-305	Software engineering	Candea	IN							*2*	*1*	*1*					4	sem A	
CS-234	Technologies for democratic society	Ford	IN	*2*	*1*	*2*										5		H	écrit
COM-309	Traitement quantique de l'information	Macris	SC							*3*	*1*						4	H	écrit
	Bloc D "SHS et MGT transversal" :															**8**			
HUM/MGT-nr	SHS : Cours à choix I selon Plan d'études SHS & MGT	Divers enseignants	CDH/CDM	**2**												2		sem A	
HUM/MGT-nr	SHS : Cours à choix II selon Plan d'études SHS & MGT	Divers enseignants	CDH/CDM				**2**									2		sem P	
HUM/MGT-nr	SHS : Cours à choix III selon Plan d'études SHS & MGT	Divers enseignants	CDH/CDM							**2**							2	sem A	
HUM/MGT-nr	SHS : Cours à choix IV selon Plan d'études SHS & MGT	Divers enseignants	CDH/CDM										**2**				2	sem P	
	Totaux:															**60**	**60**		

2021-2022 SYSTÈMES DE COMMUNICATION — Cycle Master

Code	Matières	Enseignants sous réserve de modification	Sections	Spécialisations	Semestres MA1 c	MA1 e	MA1 p	MA2 c	MA2 e	MA2 p	Crédits	Période des épreuves	Type examen
	Groupe "Core courses et options"										**72**		
	Groupe 1 "Core courses"										**min. 30**		
CS-450	Advanced algorithms	Kapralov	IN	B C D E I				**4**	**3**		7	E	écrit
COM-417	Advanced probability and applications	Lévêque	SC	B H I				**3**	**2**		6	E	écrit
COM-401	Cryptography and security	Vaudenay	SC	D E J	**4**	**2**					7	H	écrit
CS-451	Distributed algorithms	Guerraoui	SC	C E G I J	**3**	**2**	**1**				6	H	écrit
CS-423	Distributed information systems	Aberer	SC	B E J	**2**	**1**					4	H	écrit
COM-402	Information security and privacy	Hubaux/Pyrgelis	SC	B D E G	**3**	**1**	**2**				6	H	écrit
COM-404	Information theory and coding	Telatar	SC	B H I	**4**	**2**					7	H	écrit
COM-405	Mobile networks	Hubaux	SC	D E G H				**2**	**1**		4	E	écrit
COM-430	Modern digital communications: a hands-on approach	Chiurtu	SC	E F H	**2**		**2**				6	sem A	
CS-433	Machine learning	Jaggi/Flammarion	IN	B F I J	**4**	**2**					7	H	écrit
COM-500	Statistical signal and data processing through applications	Ridolfi	SC	B F H				**3**	**2**		6	E	écrit
COM-407	TCP/IP networking	Le Boudec	SC	D E G H	**2**	**2**	**2**				6	H	écrit
	Groupe 2 "Options"	**(la somme des crédits des groupes 1 et 2 doit être de 72 crédits au minimum)**											
---	Cours à option	Divers enseignants	Divers										
	Bloc "Projets et SHS" :										**18**		
COM-416	Semester project in communication systems II	divers enseignants	SC		←			**2**		→	12	sem A ou P	
HUM-nnn	SHS : introduction au projet	divers enseignants	SHS		**2**		**1**				3	sem A	
HUM-nnn	SHS : projet	divers enseignants	SHS							**3**	3	sem P	sans retrait
	Total des crédits du cycle master										**90**		

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.
Parmi les mineurs offerts par l'EPFL, la section recommande à ses étudiants les mineurs suivants :

- Biocomputing (SIN)
- Computational Science and Engineering (SMA)
- Management de la technologie et entrepreneuriat (SMTE)
- Technologies biomédicales (SMT)
- Technologies spatiales (SEL)

Le choix des cours de tous les mineurs se fait sur conseil de la section de l'étudiant et du responsable du mineur.

SYSTÈMES DE COMMUNICATION - Options

Code	Matières	Enseignants sous réserve de modification	Sections	Spécialisations	Semestres MA1 c	MA1 e	MA1 p	MA2 c	MA2 e	MA2 p	Crédits	Nbre places	Période des épreuves	Type examen	Cours biennaux donnés en
CS-470	Advanced computer architecture	Ienne	IN	A D G				3		2	6		E	écrit	
CS-440	Advanced computer graphics	Jakob	IN	F				2	1		6		sem P		
COM-501	Advanced cryptography	Vaudenay	SC	D				2	2		4		E	écrit	
CS-471	Advanced multiprocessor architecture	Falsafi	IN	A G	4						6		sem A		2021-2022
CS-523	Advanced topics on privacy enhancing technologies	Troncoso	IN	D				3	1	2	7		E	écrit	
MATH-493	Applied biostatistics	Goldstein	MA					2	2		5		sem P		
CS-401	Applied data analysis	West	IN	B	2		2				6		H	écrit	
CS-456	Artificial neural networks	Gerstner	IN					2	2		5		E	écrit	
EE-554	Automatic speech processing	Magimai Doss	EL	F	2	1					3		H	écrit	
BIO-410	Bioimage informatics	Sage/Seitz	SV	F				2		2	4		sem P	sans retrait	
BIO-465	Biological modeling of neural networks	Gerstner	IN					2	2		4		E	écrit	
EE-512	Biomedical signal processing	Vesin	EL	F	4		2				6		H	écrit	
CS-490	Business design for IT services	Wegmann	SC					3			3		E	oral	
BIO-105	Cellular biology and biochemistry for engineers	Zufferey	SV		2	2					4		H	écrit	
CS-524	Computational complexity	Göös	IN	B I	3	1					4		sem A		
CS-413	Computational photography	Süsstrunk	SC	F				2		2	5		sem P		
COM-418	Computers and Music	Prandoni	SC					2	1		4		H	écrit	
CS-442	Computer vision	Fua	IN	F				2	1		4		E	écrit	
CS-453	Concurrent algorithms	Guerraoui	SC	C G I	3	1	1				5		H	écrit	
CS-422	Database systems	Ailamaki	IN	B C G J				3	2	2	7		E	écrit	
COM-480	Data visualization	Vuillon	SC	B				2		2	4		sem P		
CS-438	Decentralized systems engineering	Ford	IN	G	2	2	2				6		H	oral	
EE-559	Deep learning	Fleuret	EL	F				2	2		4	500	E	écrit sans retrait	
CS-472	Design technologies for integrated systems	De Micheli	IN	A	3		2				6		sem A		
CS-411	Digital education & learning analytics	Dillenbourg/Jermann	IN		2		2				4		H	oral	
ENG-466	Distributed intelligent systems	Martinoli	SIE	A				2	3		5		E	oral	
COM-502	Dynamical system theory for engineers	Thiran P.	SC					2	1		4		E	écrit	
CS-473	Embedded systems	Beuchat	IN	A	2		2				4		H	oral	
CS-491	Enterprise and service-oriented architecture	Regev	SC	J				6			6		E	oral	
CS-489	Experience design	Huang	IN	F	2		4				6		sem A		
CS-550	Formal verification	Kuncak	IN	A C D	2	2	2				6		sem A		
CS-457	Geometric computing	Pauly	IN	F	3		2				6		sem A		
MATH-483	Gödel and recursivity (pas donné en 2021-2022)	Duparc	MA	I	2	2					5		H	écrit	2022-2023
CS-486	Interaction design	Pu	IN	J				2	1	1	4		sem P		
MICRO-511	Image processing I	Unser/Van De Ville	MT	F	3						3		H	écrit	
MICRO-512	Image processing II	Unser/Van de Ville/Liebling/Sage	MT	F				3			3		E	écrit	
CS-487	Industrial automation	Tournier/Sommer	SC					2		1	3		E	oral	
CS-430	Intelligent agents (pas donné en 2021-2022)	Faltings	IN	J	3	3					6		sem A		
CS-431	Introduction to natural language processing	Chappelier/Rajman	IN	B J	2	2					4		H	écrit	
CS-526	Learning theory	Macris/Urbanke	SC					2	2		4		E	écrit	
CS-421	Machine learning for behavioral data	Käser	IN					2		2	4		E	écrit	
COM-516	Markov chains and algorithmic applications	Lévêque/Macris	SC	B I	2	2					4		H	écrit	
COM-514	Mathematical foundations of signal processing	Simeoni/Bejar/Fageot	SC	F	3	2					6		H	écrit	
COM-512	Networks out of control	Thiran P./Grossglauser	SC	B E H J				2	1		4		E	écrit	2021-2022
MATH-489	Number theory in cryptography	Jetchev	MA	D				2	2		5		E	écrit	
CS-439	Optimization for machine learning	Jaggi/Flammarion	IN					2	2	1	5		E	écrit	
COM-507	Optional project in Communication Systems	Divers enseignants	SC		2						8		sem A ou P		
CS-522	Principles of computer systems	Argyraki/Candea	SC/IN	A C D G	4						7		sem A		
MATH-467	Probabilistic method in combinatorics	Marcus	MA	I	2	2					5		H	écrit	2021-2022
CS-476	Real-time embedded systems	Beuchat	IN	A				2		2	4		sem P		
EE-511	Sensors in medical instrumentation	Aminian	EL	F				2	1		3		E	écrit	
MATH-318	Set theory	Duparc	MA					2	2		5		E	écrit	
EE-472	Smart grid technologies	Paolone / Le Boudec	EL/SC					2	1	2	5		E	écrit	
EE-593	Social media	Gillet/Vonèche	EL	J				1		1	2	45	sem P	sans retrait	
CS-412	Software security	Payer	IN	D				3	2	1	6		sem P		
MATH-486	Statistical mechanics and Gibbs measures	Friedli	MA					2	2		5		E	oral	
MATH-413	Statistics for Data Science	Ohlede	MA	B	4	2					6		H	écrit	
COM-506	Student seminar: security protocols and applications	Vaudenay	SC	D				2			3		E	écrit	
CS-448	Sublinear algorithms for big data analysis	Kapralov	IN	I	2	1					4		sem A		2021-2022
CS-410	Technology ventures in IC (pas donné en 2021-2022)	Bugnion	IN					2		2	4		sem P		
CS-458	The GC Maker Project (pas donné en 2021-2022)	Pauly	IN	F						6	6		sem P		
CS-455	Topics in theoretical computer science (pas donné en 2021-2022)	Kapralov	IN	B I	3	1					4		sem A		2022-2023
CS-444	Virtual reality	Boulic	IN	F				2	1		4		sem P		

2021-2022

SYSTÈMES DE COMMUNICATION - Spécialisations

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code		Matières	Enseignants	Sections	Crédits	Période des cours	
		Spécialisation A "COMPUTER ENGINEERING	**Responsable : Prof. P. Ienne**		**60**		
CS-420	*	Advanced compiler construction	Schinz	IN	4		P
CS-470		Advanced computer architecture	Ienne	IN	6		P
CS-471		Advanced multiprocessor architecture	Falsafi	IN	6	A	
EE-431	*	Advanced VLSI design	Burg	EL	4		P
CS-472		Design technologies for integrated systems	De Micheli	IN	6	A	
ENG-466		Distributed intelligent systems	Martinoli	SIE	5		P
CS-473		Embedded systems	Beuchat	IN	4	A	
CS-550		Formal verification	Kuncak	IN	6	A	
EE-429	*	Fundamentals of VLSI Design	Burg	EL	4	A	
EE-490(b)	*	Lab in EDA based design	Koukab/Vachoux	EL	4	A	
CS-522		Principles of computer systems	Argyraki/Candea	SC/IN	7	A	
CS-476		Real-time embedded systems	Beuchat	IN	4		P
		Spécialisation B "DATA ANALYTICS"	**Responsable : Prof. M. Grossglauser/Prof. P. Thiran**		**91**		
CS-450		Advanced algorithms	Kapralov	IN	7		P
COM-417		Advanced probability and aplications	Lévêque	SC	6		P
CS-401		Applied data analysis	West	IN	6	A	
CS-524		Computational complexity	Göös	IN	4	A	
CS-422		Database systems	Ailamaki	IN	7		P
COM-480		Data visualization	Vuillon	SC	4		P
CS-423		Distributed information systems	Aberer	SC	4	A	
COM-404		Information theory and coding	Telatar	SC	7	A	
COM-402		Information security and privacy	Hubaux/Pyrgelis	SC	6	A	
COM-308	*	Internet analytics	Grossglauser	SC	5		P
CS-431		Introduction to natural language processing	Chappelier / Rajman	IN	4	A	
COM-516		Markov chains and algorithmic applications	Lévêque/Macris	SC	4	A	
COM-512		Networks out of control	Grossglauser/Thiran	SC	4		P
CS-433		Machine learning	Jaggi/Flammarion	IN	7	A	
COM-500		Statistical signal and data processing through	Ridolfi	SC	6		P
MATH-413		Statistics for Data science	Olhede	MA	6	A	
CS-455		Topics in theoretical computer science (pas donné en 2021-22)	Kapralov	IN	4	A	
		Spécialisation C "FOUNDATIONS OF SOFTWARE"	**Responsable : Prof. M. Odersky**		**46**		
CS-450		Advanced algorithms	Kapralov	IN	7		P
CS-420	*	Advanced compiler construction	Schinz	IN	4		P
CS-453		Concurrent algorithms	Guerraoui	SC	5	A	
CS-422		Database systems	Ailamaki	IN	7		P
CS-451		Distributed algorithms	Guerraoui	SC	6	A	
CS-550		Formal verification	Kuncak	IN	6	A	
CS-452	*	Foundations of software	Odersky	IN	4	A	
CS-522		Principles of computer systems	Argyraki/Candea	SC/IN	7	A	
		Spécialisation D - CYBER SECURITY - SP	**Responsable : C. Troncoso**		**82**		
CS-450		Advanced algorithms	Kapralov	IN	7		P
CS-470		Advanced computer architecture	Ienne	IN	6		P
COM-501		Advanced cryptography	Vaudenay	SC	4		P
CS-523		Advanced topics on privacy enhancing technologies	Troncoso	IN	7		P
EE-431	*	Advanced VLSI design	Burg	EL	4		P
COM-401		Cryptography and security	Vaudenay	SC	7	A	
CS-550		Formal verification	Kuncak	IN	6	A	
EE-429	*	Fundamentals of VLSI Design	Burg	EL	4	A	
COM-402		Information security and privacy	Hubaux/Pyrgelis	SC	6	A	
COM-405		Mobile networks	Hubaux	SC	4		P
MATH-489		Number theory in cryptography	Jetchev	MA	5		P
CS-522		Principles of computer systems	Argyraki/Candea	SC/IN	7	A	
CS-412		Software security	Payer	IN	6		P
COM-506		Student seminar : security protocols and applications	Vaudenay	SC	3		P
COM-407		TCP/IP networking	Le Boudec	SC	6	A	

Légende :

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

A = automne, P = printemps - 1 semestre comprend 14 semaines

2021-2022

SYSTÈMES DE COMMUNICATION - Spécialisations

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code	Matières	Enseignants	Sections	Crédits	Période des cours	
	Spécialisation E. "NETWORKING AND MOBILITY"	**Responsable : Prof. J.-Y. Le Boudec**		**50**		
CS.450	Advanced algorithms	Kapralov	IN	7		P
COM-401	Cryptography and security	Vaudenay	SC	7	A	
CS-451	Distributed algorithms	Guerraoui	SC	6	A	
CS-423	Distributed information systems	Aberer	SC	4	A	
COM-402	Information security and privacy	Hubaux/Pyrgelis	SC	6	A	
COM-405	Mobile networks	Hubaux	SC	4		P
COM-430	Modern digital communication : A hands-on approach	Chiurtu	SC	6	A	
COM-512	Networks out of control	Thiran P./Grossglauser	SC	4		P
COM-407	TCP/IP networking	Le Boudec	SC	6	A	
	Spécialisation F. "SIGNALS, IMAGES, AND INTERFACES"	**Responsables : Prof. M. Pauly**		**94**		
CS-440	Advanced computer graphics	Jakob	IN	6		P
EE-554	Automatic speech processing	Magimai Doss	EL	3	A	
BIO-410	Bioimage informatics	Sage/Seitz	SV	4		P
EE-512	Biomedical signal processing	Vesin	EL	6	A	
CS-413	Computational photography	Süsstrunk	SC	5		P
CS-442	Computer vision	Fua	IN	4		P
EE-559	Deep learning	Fleuret	EL	4		P
CS-489	Experience design	Huang	IN	6	A	
CS-457	Geometric computing	Pauly	IN	6	A	
MICRO-511	Image processing I	Unser/Van De Ville	MT	3	A	
MICRO-512	Image processing II	Unser/Van de Ville/Liebling/Sage	MT	3		P
CS-341 *	Introduction to computer graphics (pas donné en 2021-22)	Pauly	IN	6		P
COM-514	Mathematical foundations of signal processing	Simeoni/Bejar Haro/Fageot	SC	6	A	
COM-430	Modern digital communication : A hands-on approach	Chiurtu	SC	6	A	
CS-433	Machine learning	Jaggi/Flammarion	IN	7	A	
EE-511	Sensors in medical instrumentation	Aminian	EL	3		P
COM-500	Statistical signal and data processing through applications	Ridolfi	SC	6		P
CS-458	The GC Maket Project (pas donné en 2021-22)	Pauly	IN	6		P
CS-444	Virtual reality	Boulic	IN	4		P
	Spécialisation G "SOFTWARE SYSTEMS"	**Responsable : Prof. G. Candea**		**67**		
CS-420 *	Advanced compiler construction	Schinz	IN	4		P
CS-470	Advanced computer architecture	Ienne	IN	6		P
CS-471	Advanced multiprocessor architecture	Falsafi	IN	6	A	
CS-453	Concurrent algorithms	Guerraoui	SC	5	A	
CS-422	Database systems	Ailamaki	IN	7		P
CS-438	Decentralized systems engineering	Ford	IN	6	A	
CS-451	Distributed algorithms	Guerraoui	SC	6	A	
CS-452 *	Foundations of software	Odersky	IN	4	A	
COM-402	Information security and privacy	Hubaux/Pyrgelis	SC	6	A	
COM-405	Mobile networks	Hubaux	SC	4		P
CS-522	Principles of computer systems	Argyraki/Candea	SC/IN	7	A	
COM-407	TCP/IP networking	Le Boudec	SC	6	A	
	Spécialisation H. "WIRELESS COMMUNICATIONS"	**Responsable : Prof. E. Telatar**		**42**		
COM-417	Advanced probability and applications	Lévêque	SC	6		P
COM-404	Information theory and coding	Telatar	SC	7	A	
COM-405	Mobile networks	Hubaux	SC	4		P
COM-430	Modern digital communication : A hands-on approach	Chiurtu	SC	6	A	
COM-512	Networks out of control	Thiran P./Grossglauser	SC	4		P
EE-345 *	Rayonnement et antennes	Skrivervik	EL	3	A	
COM-500	Statistical signal and data processing through applications	Ridolfi	SC	6		P
COM-407	TCP/IP networking	Le Boudec	SC	6	A	

Légende :

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

A = automne, P = printemps - 1 semestre comprend 14 semaines

2021-2022

Spécialisations

Les enseignants, les crédits et la période des cours sont indiqués sous réserve de modification.

Code	Matières	Enseignants	Sections	Crédits	Période des cours	
	Spécialisation I. "COMPUTER SCIENCE THEORY"	**Responsable : Prof. O. Svensson**		**85**		
CS.450	Advanced algorithms	Kapralov	IN	7		P
COM-417	Advanced probability and aplications	Lévêque	SC	6		P
MAT-460 *	Combinatorial optimization	Eisenbrand	MA	5	A	
CS-524	Computational complexity	Göös	IN	4	A	
MATH-472 *	Computational finance	Glau/Pulido	MA	5	A	
CS.453	Concurrent algorithms	Guerraoui	SC	5	A	
MATH-461 *	Convexity	Eisenbrand	MA	5	A	
CS-451	Distributed algorithms	Guerraoui	SC	6	A	
MATH-483	Gödel and recursivity (pas donné en 2021-22)	Duparc	MA	5	A	
COM-404	Information theory and coding	Telatar	SC	7	A	
COM-300 *	Modèles stochastiques pour les communications	Thiran	SC	6	A	
CS-433	Machine learning	Jaggi/Flammarion	IN	7	A	
MATH-467	Probabilistic method in combinatorics	Pach	MA	5	A	
COM-516	Markov chains and algorithmic applications	Lévêque/Macris	SC	4	A	
CS-448	Sublinear algorithms for big data analysis	Kapralov	IN	4	A	
CS-455	Topics in theoretical computer science (pas donné en 2021-22)	Kapralov	IN	4	A	
	Spécialisation J. "INTERNET INFORMATION SYSTEMS"	**Responsable : Prof. B. Faltings et Prof. K. Aberer**		**57**		
COM-401	Cryptography and security	Vaudenay	SC	7	A	
CS-422	Database systems	Ailamaki	IN	7		P
CS-451	Distributed algorithms	Guerraoui	SC	6	A	
CS-423	Distributed information systems	Aberer	SC	4	A	
CS-491	Enterprise and service-oriented architecture	Wegmann	SC	6		P
CS-486	Interaction design	Pu	IN	4		P
CS-430	Intelligent agents (pas donné en 2021-22)	Faltings	IN	6	A	
CS-431	Introduction to natural language processing	Chappelier/Rajman	IN	4	A	
COM-512	Networks out of control	Thiran P./Grossglauser	SC	4		P
CS-433	Machine learning	Jaggi/Flammarion	IN	7	A	
EE-593	Social Media	Gillet/Vonèche	EL	2		P

Légende :

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

A = automne, P = printemps - 1 semestre comprend 14 semaines

RÈGLEMENT D'APPLICATION DU CONTRÔLE DES ÉTUDES DE LA SECTION DE SYSTÈMES DE COMMUNICATION
pour l'année académique 2021-2022
du 26 mai 2021

La direction de l'École polytechnique fédérale de Lausanne

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 de la section de systèmes de communication qui se rapportent à l'année académique 2021-2022.

Art. 2 – Étapes de formation

1. Le bachelor est composé de deux étapes successives de formation :
- le cycle propédeutique 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 informatique.
- le cycle bachelor s'étendant sur 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 semaines à l'EPFL ou de 25 semaines hors EPFL (industrie ou autre haute école) et dont la réussite se traduit par l'acquisition de 30 crédits. Il est placé sous la responsabilité d'un professeur ou MER affilié à la section de systèmes de communication ou d'informatique.

Art 3 – Sessions d'examen

1. Les branches d'examen sont examinées par écrit ou par oral pendant les sessions d'hiver ou d'été. Elles sont mentionnées dans le plan d'études avec la mention H ou E.

2. Les branches de semestre sont examinées pendant le semestre d'automne ou le semestre de printemps. Elles sont mentionnées dans le plan d'études avec la mention sem A ou sem P.

3. Une branche annuelle, c'est à dire dont l'intitulé tient sur une seule ligne dans le plan d'étude, est examinée globalement pendant la session d'été (E).

4 Pour les branches de session, la forme écrite ou orale de l'examen indiquée pour la session peut être complétée par des contrôles de connaissances écrits ou oraux durant le semestre, selon indications de l'enseignant.

Chapitre 1 : Cycle propédeutique

Art. 4 - Examen propédeutique

1 L'examen propédeutique comprend des branches « Polytechniques » pour 31 coefficients et des branches « Spécifiques » pour 29 coefficients, distribuées indifféremment sur deux blocs.

2 Le premier bloc de branches correspond à 38 coefficients et le second bloc de branches correspond à 22.

3 L'examen propédeutique est réussi lorsque :
– l'étudiant a obtenu, à l'issue de la session d'hiver, une moyenne égale ou supérieure à 3.50 dans le premier bloc, condition pour entrer au semestre de printemps, et
– qu'il a obtenu, à l'issue de la session d'été, une moyenne égale ou supérieure à 4.00 dans chacun des deux blocs, condition pour entrer au cycle bachelor.

4 L'étudiant qui échoue l'examen propédeutique ne sera pas autorisé l'année suivante à répéter les branches de semestre pour lesquelles il a obtenu une note égale ou supérieure à 4.00.

Chapitre 2 : Cycle bachelor

Art. 5 - Organisation

1 Les enseignements du bachelor sont répartis en cinq blocs, le groupe « projet », le groupe « options » et le bloc transversal SHS.

2 Le groupe « options » se compose de toutes les branches à option figurant dans la liste du plan d'études de 2ème année et 3ème année. 35 crédits doivent être obtenus individuellement dans le groupe « options », dont 12 crédits dans les options de 2ème année. Les crédits pris en supplément des 12 crédits exigés de 2ème année peuvent être validés comme crédits à options de 3ème année.

3 En 3ème année, des cours comptant pour un maximum de 10 crédits au total peuvent être choisis en dehors de la liste du plan d'études. Les cours pris en dehors de cette liste doivent être acceptés préalablement par le directeur de la section.

4 L'étudiant reste soumis au plan d'études du cycle Bachelor en vigueur lors de son entrée en 2ème année.

Art. 7 - Examen de 2ème année

1 Les **11 crédits** du plan d'études sont obtenus lorsque le bloc A est réussi.

2 Les **17 crédits** du plan d'études sont obtenus lorsque le bloc B est réussi.

3 Les **16 crédits** du plan d'études sont obtenus lorsque le bloc C est réussi.

4 Les **12 crédits de 2ème année** du groupe « options » s'acquièrent de façon indépendante, par réussite individuelle de chaque branche.

Art. 8 - Examen de 3e année

1 Les **18 crédits** du plan d'études sont obtenus lorsque le bloc D est réussi.

2 Les **7 crédits** du plan d'études sont obtenus lorsque le bloc E est réussi.

3 Les **8 crédits** du groupe « projet » s'acquièrent de façon indépendante, par réussite individuelle du projet.

4 Les **23 crédits de 3ème année** du groupe « options » s'acquièrent de façon indépendante, par réussite individuelle de chaque branche.

Art. 9 - Examen de 2e et 3e années

Le bloc « SHS transversal » est réussi lorsque les **8 crédits** du plan d'études sont obtenus.

Chapitre 3 : Cycle master

Art. 10 – Conditions d'admission

1. Les étudiants issus du Bachelor en Systèmes de communication et en Informatique sont admis automatiquement.

2. Les étudiants issus du Bachelor en Informatique qui n'auront pas suivi les cours prérequis durant leur cycle Bachelor devront les prendre en parallèle à leur cycle Master.

3. Pour les autres étudiants, l'admission s'effectue sur dossier.

Art. 11 - Organisation

1. Les enseignements du cycle master sont répartis en un bloc « Projet + SHS » et deux groupes dont les crédits doivent être obtenus de façon indépendante. Ils peuvent donner lieu à l'obtention d'une spécialisation ou d'un mineur.

2. Le bloc « Projets et SHS » est composé d'un projet de 12 crédits et de l'enseignement SHS.

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

4. 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 obtenus dans le groupe 1 « Core courses » ;
   - d'un projet optionnel de 8 crédits suivant l'alinéa 5 ;
   - de cours hors plan d'études suivant l'alinéa 6 ;
   - de cours liés à une spécialisation ou un mineur suivant l'art. 13.

5. Le projet du bloc « Projets et SHS » et le projet optionnel du groupe 2 ne peuvent être effectués dans le même semestre.

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

Art. 12 - Examen du cycle master

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

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

3. Le groupe 1 « Core courses » est réussi lorsqu'**au moins 30 crédits** sont obtenus.

Art. 13 - Enseignement SHS

Les deux branches SHS donnent chacune lieu à 3 crédits. L'enseignement du semestre d'automne introduit à la réalisation du projet du semestre de printemps. Pour autant qu'il considère que le motif est justifié, le Collège des Humanités peut déroger à cette organisation. Il peut également autoriser à ce qu'un étudiant réalise son projet sur un semestre qui ne suit pas immédiatement celui dans lequel a lieu l'enseignement d'introduction.

Art. 14 – Mineurs et spécialisations

1. Afin d'approfondir un aspect particulier de sa formation ou de développer des interfaces avec d'autres sections, 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. Le choix des cours qui composent un mineur se fait avec la section de systèmes de communication et avec le responsable du mineur. Les mineurs « Data Science », « Informatique », « Cyber Security » et « Systèmes de Communication » ne peuvent pas être choisis.

3. Le choix des cours qui composent une spécialisation est soumis, pour concertation à la section de systèmes de communication.

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

5. L'étudiant qui choisit une spécialisation dans la liste figurant dans le plan d'études s'inscrit au plus tard au début du troisième semestre des études de master.

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

Chapitre 4 : Stage et Projet de master

Art. 15 – Stage d'ingénieur

1. Les étudiants commençant leur cycle master doivent effectuer un stage d'ingénieur durant leur master :
- soit un stage d'été de minimum 8 semaines u
- soit un stage de minimum 6 mois en entreprise (en statut stage durant un semestre). Durant la période du COVID-19, la durée du stage peut être adaptée.
- soit un Projet de Master de 25 semaines en entreprise (valide le stage et le Projet de Master)

2. En règle générale, pour les étudiants issus du Bachelor IC, le stage peut être effectué dès le 2ème semestre du cycle master, mais avant le projet de master. Sur demande de l'étudiant, la section peut l'autoriser à effectuer son stage avant ou pendant le 1er semestre du cycle Master.

3 L'étudiant ne peut pas faire de cours/projet en parallèle à son stage.

4. Le responsable du stage de la section évalue le stage, par l'appréciation « réussi » ou « non réussi ». Sa réussite sera une condition pour l'admission au projet de master. En cas de non réussite, il pourra 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'une directive interne à la section.

Chapitre 5 : Spécialisation Enseignement

Art. 16 – Spécialisation Enseignement

1. Les étudiants en Master Systèmes de communication ont la possibilité de suivre une spécialisation en informatique pour l'enseignement.

2. L'étudiant admis à cette spécialisation ne peut pas suivre de spécialisation ou de mineur. Le plan d'études est modifié comme suit : (i) Un nouveau groupe de 30 ECTS de cours à la HEP Vaud est rajouté et le nombre de ECTS du Cycle Master passe de 60 à 30 ECTS ; (ii) les cours SHS sont remplacés par un cours à la HEP Vaud ; (iii) le Projet de Master peut s'étaler sur deux semestres et commencer après que l'étudiant a complété le bloc « Projets et SHS » et le groupe « Core courses » ; (iv) la durée maximale des études ne peut pas dépasser 8 semestres.

3. Au moins 50 ECTS doivent avoir été obtenus pour débuter la spécialisation.

Art. 17 – Procédure d'admission

1. L'admission à cette spécialisation n'est pas automatique. Pour être admis à la spécialisation, 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'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.

Chapitre 6 : Mobilité

Art. 18 – Périodes de mobilité autorisées

Les étudiants de la section des systèmes de communication peuvent effectuer un séjour de mobilité en 3ème année de bachelor et/ou dans le cadre du projet de master.

Art. 19 - Conditions

1. Pour une mobilité en 3ème année de bachelor, l'étudiant doit avoir réussi l'examen propédeutique avec une moyenne minimale de 4,5 et ne pas avoir de retard dans l'acquisition des 60 crédits de la 2ème année de bachelor.

2 Pour une mobilité au projet de master, l'étudiant peut-être admis conditionnellement s'il n'a pas plus de 8 crédits manquants au cycle master.

3. Des conditions spécifiques existant en fonction des destinations, l'accord du délégué à la mobilité est nécessaire pour partir en séjour de mobilité.

Au nom de la direction de l'EPFL

Le président, M. Vetterli
Le vice-président académique, J. S. Hesthaven

Lausanne, le 26 mai 2021

CS-101 Advanced information, computation, communication I

Aberer Karl

Cursus	Sem.	Type
Cyber security minor	H	Opt.
Informatique	BA1	Obl.
Systèmes de communication	BA1	Obl.

Language	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, basic generating functions.
VII. Basic probability: events, independence, random variables, Bayes' theorem.
VIII. Structure of sets: relations, equivalence relations, power set.
IX. (time permitting) Elementary graph theory: graphs, Euler and Hamilton paths, Dijkstra's algorithm, spanning trees.

Keywords

Propositional logic, counting, complexity, big-O, number representations, sets, matrices, modular arithmetic, induction, basic probabilities, Bayes theorem, combinatorial analysis, recurrences, generating functions, countability, graph theory.

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

Final exam (100%), mostly (and possibly exclusively) multiple choice

Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	A list of students assistants and their contact data will be made available on the moodle page for this course, along with an assignment of each registered student to one of the student assistants. If you have a question, first contact the student assistant assigned to you. If that does not help, contact one of the teaching assistants (Negar Foroutan, Banaei Mohammadreza). Furthermore, you are always welcome to drop me an email (karl.aberer@epfl.ch) for any type of question related to this course or your study at EPFL. Never hesitate to ask questions before, during or after the lectures!

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

- http://will be provided later

COM-102

Advanced information, computation, communication II

Gastpar Michael C.

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

Language	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 and probability. Entropy per symbol. Source coding.

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

III. Protecting information. A few finite fields. Linear speaces. Hamming distance. Linear codes. Reed-Solomon codes.

Keywords

Shannon's entropy
Linear codes
Reed-Solomon codes
Number theory
Asymmetric Cryptography, RSA

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

- http://moodle.epfl.ch/course/view.php?id=851

MATH-111(g)

Algèbre linéaire

Zuleta Estrugo José Luis

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	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

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 interprêter 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

Méthode d'évaluation

examen écrit

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Bibliographie

Algèbre linéaire et applications, David C. Lay, 5e edition, editeur: Pearson, ISBN: 978-2-7613-9109-2

Ressources en bibliothèque

- Algèbre linéaire et applications / Lay

Préparation pour

Analyse II

MATH-111(en) **Algèbre linéaire (anglais)**

Maddocks John H.

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.

Language	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	**230**

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

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	No

Resources

Bibliography

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

Ressources en bibliothèque

- Linear Algebra and its Applications / Lay

Prerequisite for

Analysis II, III and IV, Numerical Analysis Statistics

EPFL

MATH-111(pi) Algèbre linéaire (classe inversée)

Urech Christian Lucius

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	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	**184**

Remarque

Cours en classe inversée, merci de consulter http://go.epfl.ch/algebre-pilote 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	Oui

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

- http://moodle.epfl.ch/course/search.php?search=MATH-111%28pi%29
- http://moodle.epfl.ch/course/view.php?id=15414

Préparation pour

Suite des études en ingénierie et sciences.

MATH-101(g)

Analyse I

Chizat Lénaïc

Cursus	Sem.	Type
Chimie et génie chimique	BA1	Obl.
Science et génie des matériaux	BA1	Obl.
Systèmes de communication	BA1	Obl.

Langue	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	Non
Autres	Tutorat des exercices autres mesures à définir

Ressources

Bibliographie

Jacques Douchet and Bruno Zwahlen: Calcul différentiel et intégral. Volume 1. PPUR, 2016.

Ressources en bibliothèque

- (version électronique)
- Calcul différentiel et intégral / Douchet & Zwahlen

EPFL

MATH-101(de) **Analyse I (allemand)**

Kressner Daniel

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	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	**60**

Résumé

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

Contenu

- Mathematisches Begründen, Beweisen, und Argumentieren
- 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.

MATH-101(en) **Analyse I (anglais)**

Svaldi Roberto

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.

Language	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	**239**

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

MATH-101(pi) **Analyse I (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	Opt.
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	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	**184**

Remarque

Cours en classe inversée, merci de consulter https://go.epfl.ch/analyse-pilote 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 donné 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 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	Oui
Autres	Tutorat des exercices. Portail Moodle.

Ressources

Ressources en bibliothèque

-

Liens Moodle

- https://moodle.epfl.ch/course/view.php?id=16766

Préparation pour

Analyse II

EPFL

MATH-106(e) Analyse II

Lachowska Anna

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

Langue	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

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.

MATH-106(en) **Analyse II (anglais)**

Marcus Adam W.

Cursus	Sem.	Type
Chimie et génie chimique	BA2	Obl.
Génie civil	BA2	Obl.
Génie mécanique	BA2	Obl.
Génie électrique et électronique	BA2	Obl.
Informatique	BA2	Obl.
Ingénierie des sciences du vivant	BA2	Obl.
Microtechnique	BA2	Obl.
Science et génie des matériaux	BA2	Obl.
Sciences et ingénierie de l'environnement	BA2	Obl.
Systèmes de communication	BA2	Obl.

Language	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

Teaching methods

Ex cathedra lectures, exercises sessions in the classroom.

Assessment methods

Written exam

Supervision

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

Resources

Bibliography

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

Websites

- http://mcss.epfl.ch/page-105207-en.html

CS-173 Digital system design

Kluter Ties Jan Henderikus

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

Language	English
Coefficient	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Practical work	2 weekly
Number of positions	

Summary

The goal is to familiarize the students with the hardware components of computing systems, and to teach the modern methods of analysis and synthesis of combinational and sequential systems, with the assistance of high-level languages such as VHDL.

Content

1. Analog versus digital, logic: the principles and the operators.
2. Boolean algebra, combinational functions, and Karnaugh diagrams.
3. Sequential functions and their three representatives.
4. Coding, how do we interpret those bits.
5. Finite state machines.
6. Real gates and technology.
7. Programmable logic and their application.
8. Describing systems at a higher level by using a hardware description language (VHDL)
9. Application of the learned theory by practical problems.

Keywords

Digital system design, logic gates, Boolean algebra, gates, flip-flops, latches, FPGA, CPLD, FSM, coding, VHDL

Learning Prerequisites

Required courses

None

Recommended courses

None

Learning Outcomes

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

- Identify sequential and combinational logic functions
- Recognize logic functions, gates, latches, and flipflops
- Describe simple digital systems in VHDL
- Analyze digital systems either described in VHDL or implemented with gates
- Implement a digital system from a problem description

- Solve boolean equations and number system problems
- Design a complete digital system that runs on FPGA
- Detect differences between the theory and the practical application

Transversal skills

- Use a work methodology appropriate to the task.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

Course ex cathedra, exercises and practical laboratory projects

Expected student activities

The student must attend the course, prepare and solve the exercises, prepare and carry out the laboratory projects

Assessment methods

Midterm test (40%)
Final test (40 %)
Midterm practical project (10%)
Final practical project (10%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

W. J. Dally and R. C. Harting,Digital design : A systems approach, Cambridge University Press, 2012J. Wakerly,Digital design (4th edition), Prentice Hall, 2005P. J. Ashenden,The student's guide to VHDL (2nd edition), Morgan Kaufmann, 2008C. Maxfield,Bebop to the boolean boogie: An unconventional guide to electronics (3rdedition), Newnes, 2008

Ressources en bibliothèque

- C. Maxfield, Bebop to the boolean boogie: An unconventional guide to electronics (3rd ed)
- W. J. Dally and R. C. Harting, Digital design : A systems approach
- J. Wakerly, Digital design (4th ed)
- P. J. Ashenden, The student's guide to VHDL (2nd ed)

Notes/Handbook

All material is available on moodle including:

- Slides
- Theory booklet

- Exercises
- Old exams

Prerequisite for

Computer Architecture (CS-208)
Systems-on-Chip Architecture (CS-209)

CS-107

Introduction à la programmation

Sam Jamila

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

Langue	français
Coefficient	5
Session	Hiver
Semestre	Automne
Examen	Pendant le semestre
Charge	150h
Semaines	14
Heures	**5 hebdo**
Cours	2 hebdo
Exercices	3 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 familisarier avec un environnement informatique (station de travail sous UNIX)

Contenu

• Introduction à l'environnement UNIX (connection, multi-fenêtrage, édition de textes, email, ...), éléments de base du fonctionnement d'un système informatique et 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, 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

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 (20%)
3- Mini-projet 2 (40%)
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

Préparation pour

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

EPFL

PHYS-101(g) **Physique générale : mécanique**

Brantut Jean-Philippe

Cursus	Sem.	Type
Systèmes de communication	BA1	Obl.

Langue	français
Coefficient	6
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	180h
Semaines	14
Heures	**6 hebdo**
Cours	3 hebdo
Exercices	3 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://moodle.epfl.ch/course/view.php?id=15451

Préparation pour

Physique générale II

PHYS-101(en) **Physique générale : mécanique (anglais)**

Manley Suliana

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.

Language	English
Coefficient	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	3 weekly
Number of positions	**308**

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 may contain, but not exclusively, the following elements :

Mechanics

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

Dynamics of the point mass and solid body
Momentum, Newton's laws, fundamental forces, empirical forces and constraints. Oscillatory motion, 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", see on the left the hyperlinks and the book, indicative of 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:

- Develop a know-how to solve a problem
- Structure models in terms of differentials equations
- Apply simplifying assumptions to describe an experience

- Estimate orders of magnitude
- Distinguish the theoretical models describing Natura
- Contextualise theoretical models in every day life
- Formulate a physical model

Transversal skills

- Use a work methodology appropriate to the task.

Teaching methods

Lectures + 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

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

Prerequisite for

General physics II

EPFL

CS-108

Pratique de la programmation orientée-objet

Schinz Michel

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

Langue	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

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

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

Sites web

- https://cs108.epfl.ch/

MATH-310 **Algebra**

Lachowska Anna

Cursus	Sem.	Type
Chimie	BA5	Opt.
Cyber security minor	H	Opt.
HES -SC	H	Obl.
Informatique	BA5	Opt.
Systèmes de communication	BA5	Obl.

Language	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Study basic concepts of modern algebra: groups, rings, fields.

Content

- Algebraic structures: sets, groups, rings, fields.
- Groups. Subgroups. Homomorphisms of groups, normal subgroups, quotients. Cyclic groups, symmetric groups. Classification of finite abelian groups.
- Rings. Homomorphisms of rings. Ideals, principal, prime and maximal ideals, principal ideal domains. Quotient rings. The Chinese remainder theorem.
- Examples of rings. Integers. basic properties. Euler's and Fermat's theorems. Polynomial rings. GCD, unique factorization.
- Fields. Finite fields. Characteristic of a field.

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, unique principal ideal domain, Euler's totient function, field, finite field, characteristic of a field.

Learning Prerequisites

Required courses

Linear Algebra I, Analyse I

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

Teaching methods

Lectures and exercise sessions

Assessment methods

Three short in-class tests (15% of the grade)
Written exam (85 % of the grade)

Supervision

Office hours	No
Assistants	Yes
Forum	No

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

Notes/Handbook

Complete lecture notes will be available in PDF

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15441

CS-250

Algorithms

Svensson Ola Nils Anders

Cursus	Sem.	Type
Cyber security minor	H	Opt.
HES - IN	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Mathématiques	BA5	Opt.
Mineur en Data science	H	Opt.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Systèmes de communication	BA3	Obl.

Language	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

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, linked lists, trees, heaps, hashing, graphs.

Design of algorithms by induction

- Divide-and-conquer algorithms, dynamic programming.

Greedy Algorithms

• Spanning tree and shortest path algortihms

Sorting and searching

- Merge sort, bucket sort, quicksort, heapsort, binary search.

Graphs algorithms and data structures

- Graphs traversals, shortest paths, spanning trees, transitive closure, decompostitions, matching, network flows.

Keywords

algorithms, data structures, efficiency, problem solving

Learning Prerequisites

Recommended courses

Advanced ICC I

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, heaps, 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 based on known methods
- Prove the correctness of an algorithm

Teaching methods

Ex cathedra lecture, exercises in classroom

Assessment methods

Continuous assessment with final exam.

Resources

Bibliography

Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein: *Introduction to algorithms*, Third Edition, MIT Press, 2009.

Ressources en bibliothèque

- Introduction to algorithms / Cormen

EPFL

MATH-203(c) **Analyse III**

Tione Riccardo, Widmayer Klaus Martin

Cursus	Sem.	Type
Chimie	BA5	Opt.
HES - SIE	H	Obl.
HES -SC	H	Obl.
Sciences et ingénierie de l'environnement	BA3	Obl.
Systèmes de communication	BA3	Obl.

Langue	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 étudie les concepts fondamentaux de l'analyse vectorielle et l'analyse de Fourier en vue de leur utilisation pour résoudre des problèmes pluridisciplinaires d'ingénierie scientifique.

Contenu

Analyse vectorielle
Les opérateurs gradient, rotationnel, divergence et laplacien. Intégrales curvilignes et intégrales de surfaces. Champs vectoriels et potentiels. Théorèmes de Green, de la divergence et de Stokes.
Analyse de Fourier
Séries de Fourier. Identité de Parceval. Transformées de Fourier. Identité de Plancherel. Utilisations et applications.

Compétences requises

Cours prérequis obligatoires

Analyse I, Analyse II, Algèbre linéaire.

Acquis de formation

- Comprendre et maîtriser les notions, les concepts et les méthodes étudiés au cours.
- Comprendre et maîtriser les notions, les concepts et les méthodes pratiqués dans les séries d'exercices.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Service de cours virtuels (VDI)

Non

Bibliographie

B. Dacorogna et C. Tanteri, *Analyse avancée pour ingénieurs,* PPUR 2018.

Ressources en bibliothèque

- Analyse avancée pour ingénieurs / Dacorogna

Préparation pour

Analyse IV.

EPFL

MATH-207(d) **Analyse IV**

Vacat .

Cursus	Sem.	Type
Chimie	BA6	Opt.
Génie civil	BA6	Opt.
HES - SIE	E	Obl.
HES -SC	E	Obl.
Informatique	BA4	Opt.
Sciences et ingénierie de l'environnement	BA4	Obl.
Systèmes de communication	BA4	Obl.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	2 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Le cours étudie les concepts fondamentaux de l'analyse complexe et de l'analyse de Laplace en vue de leur utilisation pour résoudre des problèmes pluridisciplinaires d'ingénierie scientifique.

Contenu

Analyse complexe
Définitions et exemples de fonctions complexes. Fonctions holomorphes. Equations de Cauchy-Riemann. Intégrales complexes. Formule de Cauchy. Séries de Laurent. Théorème des résidus**.**
Analyse de Laplace
Transformées de Laplace. Applications à des équations différentielles ordinaires. Applications à des équations aux dérivées partielles.

Compétences requises

Cours prérequis obligatoires

Algèbre linéaire, Analyse I, Analyse II, Analyse III.

Acquis de formation

- Comprendre et maîtriser les notions, les concepts et les méthodes étudiés au cours.
- Comprendre et maîtriser les notions, les concepts et les méthodes pratiqués dans les séries d'exercices.

Méthode d'évaluation

Examen écrit.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Non

Ressources

Service de cours virtuels (VDI)

Non

Bibliographie

B. Dacorogna et C. Tanteri, Analyse avancée pour ingénieurs, PPUR 2018.

Ressources en bibliothèque

- Analyse avancée pour Ingénieurs / Dacorogna

CS-308

Calcul quantique

Macris Nicolas

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	4
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	3 hebdo
Exercices	1 hebdo
Nombre de places	

Remarque

Cours indépendant de "Traitement quantique de l'information" (COM-309)

Résumé

La miniaturisation des ordinateurs conduit à réviser les paradigmes du calcul classique pour développer des modèles de calcul quantique. Le cours introduit les notions de bit quantique, portes logiques et circuits quantiques, traite les principaux algorithmes quantiques, et les machines IBM Q.

Contenu

Intrduction au calcul quantique
- Calcul classique: modèle des circuits classiques, calcul réversible.
- Bits quantiques, espace de Hilbert de N qubits, transformations unitaires et portes logiques élémentaires, postulat de la mesure.
- Modèle des circuits quantiques, portes universelles.
- Problème de Deutsch et Josza.

Algorithmes de base
- Sous espace vectoriel cache et algorithme de Simon.
- Intermède mathématique: factorisation d'un entier et période de fonctions discrètes. Notions sur les fractions continuées.
- Transformée de Fourier quantique et algorithme de recherche de la période d'une fonction discrète.
- Algorithme de factorisation de Shor.
- Algorithme de Grover pour la recherche dans une base de donnée.

Intrication (sujet a choix et facultatif)
- Etats intriques et circuits associés.
- Protocoles avec opérations locales quantiques + communication classique.
- Protocole de distillation et mesure de l'intrication.

Codage (sujet a choix et facultatif)
- Modèles de bruit et erreurs dans les états quantiques**.**
- Code correcteurs de Shor et Steane.
- Codes stabilisateurs.

Mots-clés

Calcul quantique, circuits quantiques, portes universelles, transformée de Fourier quantique, algorithme de Shor, Grover, intrication, codes quantiques.

Compétences requises

Cours prérequis obligatoires

Algèbre linéaire.

Concepts importants à maîtriser
Matrices, valeurs et vecteurs propres, produit scalaire, nombre complexes.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Expliquer le concept d'algorithme quantique sur le modèle des circuits.
- Connaitre les portes universelles utilisées dans un circuit quantique.
- Expliquer les principaux algorithmes quantiques
- Calculer l'évolution d'un état à travers un circuit quantique
- Appliquer le postulat de la mesure
- Faire des calculs algébriques impliquant des états à plusieurs qubits en notation de Dirac
- Se familiariser avec IBM Q et Qiskit

Méthode d'enseignement

Ex-Cathedra. Exercices. Utilisation des ordinateurs quantiques IBM Q

Travail attendu

Participation au cours, exercices et utilisation des machines IBM Q

Méthode d'évaluation

mini project on IBM Q experience, graded homeworks, examen final ecrit.

Ressources

Bibliographie
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

Polycopiés
Notes de cours

Sites web

- http://ipg.epfl.ch/doku.php?id=en:courses

Préparation pour

COM-611 Quantum Information Theory and Computation

CH-160(b)

Chimie générale

Terrettaz Samuel

Cursus	Sem.	Type
Génie civil	BA1	Obl.
Informatique	BA5	Opt.
Systèmes de communication	BA5	Opt.

Langue	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

Cet enseignement vise l'acquisition des notions essentielles relatives à la structure de la matière, aux équilibres et à la réactivité chimiques. Le cours et les exercices fournissent la méthodologie permettant de résoudre par le raisonnement et le calcul des problèmes inédits de chimie générale.

Contenu

1. **Atomistique:** structure électronique des atomes, orbitales atomiques, spectroscopie, classification périodique
2. **Liaison chimique:** représentation de Lewis, règle de l'octet, liaison ionique, liaison covalente, énergie de liaison, modèle VSEPR et géométrie des molécules, orbitales moléculaires, moment dipolaire, forces de van der Waals et de London, liaisons intermoléculaires
3. **Quantités chimiques:** masse atomique, isotopes, notion de mole, formules chimiques, concentrations
4. **Réactions chimiques et stoechiométrie:** équations chimiques, réactif limitant, électrolytes, lois des gaz parfaits, pressions partielles
5. **Thermochimie:** énergie interne, premier principe de la thermodynamique, enthalpies des transformations physiques et des réactions chimiques, entropie deuxième principe, enthalpie libre
6. **Equilibres chimiques:** enthalpie libre dans un mélange, potentiel chimique et activité, quotient réactionnel, constante d'équilibre, influence dse paramètres réactionnels sur les équilibres
7. **Propriétés des solutions:** dissolution et solvatation, solubilité, lois de Raoult et de Henry, propriétés colligatives des solutions (ébullioscopie, cryoscopie, pression osmotique)
8. **Transfert de proton:** équilibres acide-base: théorie de Bronsted-Lowry, couples acide-base, constante d'ionisation, échelle de pH, calcul de pH de solutions, titrages acide-base
9. **Transfert d'électron:** électrochimie: équilibrage des équations rédox, piles électrochimiques, potentiels standard, piles et accumulateurs, équation de Nemst, loi de Faraday, électrolyse
10. **Cinétique chimique:** vitesse de réaction, lois de vitesse, molécularité et ordre d'une réaction, théorie du complexe activé, loi d'Arrhenius, catalyse

Mots-clés

Structure électronique des atomes, liaisons chimiques, stoechiométrie, thermochimie, équilibres thermodynamiques, acides et base, oxydoréduction, cinétique chimique

Méthode d'évaluation

Examen écrit

Ressources

Bibliographie

• Chimie générale / Hill

- Chimie des solutions / Hill
- Exercices de chimie générale / Comninellis

Ressources en bibliothèque

- Chimie générale / Hill
- Chimie des solutions / Hill
- Exercices de chimie générale / Comninellis

CS-208

Computer architecture I

Stojilovic Mirjana

Cursus	Sem.	Type
HES - IN	H	Obl.
Informatique	BA3	Obl.
Mineur en Informatique	H	Opt.
Systèmes de communication	BA3	Obl.

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

Summary

The course introduces the students to the basic notions of computer architecture and, in particular, to the choices of the Instruction Set Architecture and to the memory hierarchy of modern systems.

Content

- Complex digital systems in VHDL.
- Basic components of a computer.
- Instruction Set Architectures.
- Assembly-level programming.
- Multi-cycle implementation of processors.
- Caches.
- Virtual memory.

Keywords

Computer Architecture, Basic Processor Architecture,
Instructions Sets, Cache Hierarchies, Virtual Memory.

Learning Prerequisites

Required courses

Digital system desing

Important concepts to start the course

- Digital design in VHDL
- FPGA design software: Intel Quartus
- Simulation and verification of digital systems behavior: ModelSim.

Learning Outcomes

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

- Design and implement a processor at the register transfer level using logic synthesizers and simulators.

- Develop assembly language programs.
- Justify the organization of a modern memory system including cache hierarchy.
- Design and implement a cache memory.

Teaching methods

- Ex cathedra / online lectures and exercises.
- Labs on dedicated FPGA boards.

Expected student activities

- Attending the course and exercise/lab sessions (in person or online)
- Completing the lab assignments and quizzes.
- Homework: solving individually the exercises in the course exercise book.
- Participating in the discussions on the forum.

Assessment methods

Graded labs and quizzes, during the semester (30%)
Final exam, during the exam session (70%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

Yes

Bibliography

David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kauffman, 5th edition, 2013.

Ressources en bibliothèque

- Computer organization and design

Websites

- https://parsa.epfl.ch/course-info/cs208/

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=14225

Prerequisite for

Computer architecture II

EPFL

CS-209

Computer architecture II

Ienne Paolo

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Opt.

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

Summary

The course completes the introduction to computer architecture.

Content

- Inputs/Outputs and Interrupts
- Exceptions
- Computer Performance
- Pipelining
- Dynamic Scheduling
- Superscalar and VLIW Processors
- Multiprocessors and Cache Coherence

Keywords

Computer Architecture, Processor, CPU, ILP, Multiprocessors, Coherence

Learning Prerequisites

Required courses

- CS-173 (Digital System Design)
- CS-208 (Computer Architecture I)

Learning Outcomes

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

- Design a simple exception handler in assembler
- Design pipelined digital circuits at Register Transfer Level
- Optimize the performance of a processor pipeline by reordering instructions
- Explain possible solutions to the cache coherence problem

Teaching methods

Ex-cathedra courses and labs on an FPGA board.

Assessment methods

- Lab I (13%)
- Pipeline simulation (4%)
- Lab II (13%)
- Final exam (70%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kauffman, 5th edition, 2013.

Ressources en bibliothèque

- Computer organization and design

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=14153

Prerequisite for

- CS-470 (Advanced Computer Architecture)

CS-320

Computer language processing

Kuncak Viktor

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

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Practical work	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 the new web standard for portable binaries called WebAssembly (https://webassembly.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

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

Assessment methods

The grade is based on the programming, testing, documentation, and presentation of projects done on student's own laptops during the semester.
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)
Niklaus Wirth: **Compiler Construction,** neat textbook from a prominent classical authority. Freely available **http://www.ethoberon.ethz.ch/WirthPubl/CBEAll.pdf**

Ressources en bibliothèque

- Additionally, all material
- Modern compiler implementation in Java / Appel

- Compiler Construction / Wirth
- 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

Prerequisite for

Advanced compiler construction
Recommended for Foundations of software

EPFL

COM-208

Computer networks

Argyraki Katerina

Cursus	Sem.	Type
Cyber security minor	H	Opt.
HES - IN	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA3	Obl.
UNIL - Sciences forensiques	H	Opt.

Language	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 provides an introduction to computer networks. It describes the principles that underly modern network operation and illustrates them using the Internet as an example.

Content

- Overview of Internet operation (main components and protocols).
- Application layer (web, cookies, peer to peer).
- Socket programming (how to write a very simple network application).
- Transport layer (UDP, TCP, congestion control).
- Network layer (IP forwarding and basic routing).
- Data link layer (switching).
- Security (secure email, SSL, IPsec).

Keywords

- Computer networks
- Internet
- HTTP
- Peer-to-peer networks
- Sockets, TCP/IP, congestion control, routing, switching, network security.

Learning Prerequisites

Required courses

- CS 107 - Introduction to programming
- COM 101 - Advanced Information Computation Communication I

Learning Outcomes

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

- Design simple network applications.

- Choose which functions to implement at each network layer.
- Compare different network protocols.
- Perform simple network troubleshooting.
- Use simple network monitoring tools.
- Implement simple client-server applications.
- Investigate simple network attacks.
- Explain how basic Internet applications work.
- Explain how TCP/IP works.

Transversal skills

- Use both general and domain specific IT resources and tools
- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking
- Demonstrate a capacity for creativity.

Teaching methods

- Lectures
- Reading sssignments
- Homework problems
- Hands-on exercises

Expected student activities

The students are expected to:

- attend the lectures
- complete homework problems
- complete hands-on exercises
- study their notes and -- when needed -- complement by reading relevant book chapters

Assessment methods

- Final exam
- Midterm exam (online)
- Quizzes (online)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Computer Networking: A Top-Down Approach by James F. Kurose and Keith W. Ross.

Ressources en bibliothèque

- Computer Networking / Kurose

Websites

- https://sites.google.com/site/com208computernetworks/

EPFL

COM-301

Computer security

Troncoso Carmela

Cursus	Sem.	Type
Cyber security minor	H	Opt.
Informatique	BA5	Obl.
Mineur en Informatique	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA5	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	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 Operative 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

- Computer security by Dieter Gollmann
- Security Engineering by Ross Anderson
- Computer Security: Principles and Practice by Stallings and Brown

Prerequisite for

- COM-402 Information security and privacy
- CS-523 Advanced topics on privacy enhancing technologies

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	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.

Encadrement

Office hours	Oui
Assistants	Oui

Ressources

Bibliographie

1) Support de cours par R. Fleury, disponible sur Moodle.
Pour aller plus loin:
2) "Électromagnétisme", Vol. III du Traité d'électricité de l'EPFL -
3) Ramo: "Fields and Waves in Communication Electronics"

Ressources en bibliothèque

- Fields and Waves in Communication Electronics / Ramo
- Electromagnétisme / Gardiol

Liens Moodle

- http://moodle.epfl.ch/enrol/index.php?id=14046

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	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.

Encadrement

Office hours	Oui
Assistants	Oui

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

Ressources en bibliothèque
- Electromagnétisme / Gardiol
- Fields and Waves in Communication Electronics / Ramo

Polycopiés
Disponible sur Moodle.

Liens Moodle
- http://moodle.epfl.ch/enrol/index.php?id=14047

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

EE-202(b) **Electronique I**

Zysman Eytan

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

Langue	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'enseignement

En raison du COVID-19, l'enseignement, incluant cours et exercices, se fera en visioconférence. Les enregistrements des séances seront disponibles sur Moodle.

Méthode d'évaluation

Toujours en raison du COVID-19, l'évaluation se fera sous la forme d'un rapport individuel sur des problèmes de conception et les résultats devront être validés par des simulations. Le rapport sera suivi d'une interrogation orale.
Le rapport devra être remis avant les vacances de Noël et l'oral sera organisé dès la rentrée de janvier.

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.
- Développements en cours sur écran interactif ou tablet

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=13726

Préparation pour

EE-203(b)

Electronique II

Zysman Eytan

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

Langue	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

Exercices et travaux pratiques
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

Acquis de formation

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

- Concevoir des filtres actifs

- Concevoir des circuits amplificateurs
- Comparer les différentes familles logiques
- Analyser la bande passante d'une fonction électronique
- Exploiter des blocs fonctionnels de haut niveau
- Représenter la notion de temps
- Synthétiser des circuits logiques

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'enseignement

En raison du COVID-19, l'enseignement, incluant cours et exercices, se fera en visioconférence. Les enregistrements des séances seront disponibles sur Moodle.

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.
- Développements en cours sur écran interactif ou tablet

Liens Moodle

- http://moodle.epfl.ch/course/view.php?id=13727

Préparation pour

Electronique III

EPFL

EE-381

Electronique III

Zysman Eytan

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	2 hebdo
Exercices	1 hebdo
Nombre de places	

Résumé

Comparaison entre les systèmes à composants discrets et les systèmes intégrés. Introduction aux systèmes électroniques numériques et analogiques et à leur interfaçage. Analyse sous forme d'un projet d'un cahier des charges d'un système intégré mixte analogique/numérique.

Contenu

Cours

- Cellules analogiques: miroir de courant, paire différentielle, Push-pull,...
- Conversion A/N et N/A : introduction - définitions, conversion numérique/analogique, conversion analogique/numérique.
- Oscillateur et boucles à verrouillage de phase ou Phase-Locked Loops (PLL)
- Introduction aux technologies mixtes analogiques et numériques
- Techniques de conception de circuits intégrés
- Application aux ASIC analogiques/numériques

Exercices
l'étudiant analysera et simulera de nombreux blocs fonctionnels vus en cours
projet
L'étudiant fera la conception d'un petit système électronique mixte analogique et numérique et évaluera sa complexité sous forme de circuit intégré.

Mots-clés

paire différentielle, miroir de courant, structure cascod, charge active, Push-Pull, Darlington, Wilson, Widlar, Full Custom, Semi-custom, Librairie de cellules, FPGA, EPLD, PLA, ROM, Architecture de circuit intégré, Technologie des semiconducteurs, PLL, Stabilité, Oscillateur, Convertisseur incrémental, convertisseur logarithmique, convertisseur flash et semi Flash, Sigma/Delta.

Compétences requises

Cours prérequis indicatifs

Cours d'électronique de base I et II

Concepts importants à maîtriser

Automates de Moore et de Mealy.
Transformée de Laplace.

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:
- Analyser un cahier des charges en électronique
- Concevoir un système électronique
- Décrire le comportement du circuit sous forme algorithmique
- Estimer la complexité et les performances du circuit

Compétences transversales

- Fixer des objectifs et concevoir un plan d'action pour les atteindre.
- 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.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Accéder aux sources d'informations appropriées et les évaluer.
- Ecrire un rapport scientifique ou technique.
- Faire une présentation orale.

Méthode d'enseignement

- Cours ex cathedra et exercices dirigés en salle.
- Animation de séances de projet

Travail attendu

- Remise d'un rapport d'analyse de système électronique

Méthode d'évaluation

- Travail écrit
- Rapport et présentation orale du projet

Encadrement

Office hours	Non
Assistants	Non
Forum électronique	Oui

Ressources

Bibliographie

Traité de l'électronique analogique et numérique , 1, Techniques analogique et numérique, Paul Horowitz, Winfield Hill, Elektor, 2009

Ressources en bibliothèque

- Traité de l'électronique analogique et numérique (vo.1)/ Horrowitz

Polycopiés

- liste de sites approfondissant les notions vues en cours

- Diapositives du cours
- Diapositives commentées
- Exercices et corrigés.
- Développements en cours sur Tablet
- Cahier des charges du projet

CS-210

Functional programming

Kuncak Viktor, Odersky Martin

Cursus	Sem.	Type
HES - IN	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Mineur en Data science	H	Opt.
Mineur en Informatique	H	Opt.
Systèmes de communication	BA3	Opt.

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

Summary

Understanding of the principles and applications of declaratative programming, the fundamental models of program execution, application of fundamental methods of program composition,meta-programming through the construction of interpreters and advanced programming techniques.

Content

Introduction to programming in Scala
Functions and Evaluation
Higher-Order Functions
Data and Abstraction
Types and pattern matching
Lists
Collections
Lazy evaluation
For expressions, generators and monads
Functions and State
Lambda calculus and Lisp
Interpreting Functional Languages

Learning Prerequisites

Required courses

CS-107 Introduction to programming
CS-108 Practice of object-oriented programming

Learning Outcomes

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

- Create functional programs
- Design robust and readable software
- Formalize program correctness
- Interpret programs automatically
- Prove correctness using induction
- Construct software
- Demonstrate a capacity for creativity.
- Use a work methodology appropriate to the task.

- Set objectives and design an action plan to reach those objectives
- Give feedback (critique) in an appropriate fashion.

Transversal skills

- Demonstrate a capacity for creativity.
- Use a work methodology appropriate to the task.
- Set objectives and design an action plan to reach those objectives.
- Give feedback (critique) in an appropriate fashion.

Teaching methods

MOOC. Ex Cathedra. Exercises and projects

Assessment methods

- 70% exam during the semester
- 25% programming lab assignments during the semester
- 5% written exercises during the semester

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Abelson/Sussman : Structure and Interpretation of Computer Programs, MIT Press
Martin Odersky, Lex Spoon, and Bill Venners: Programming in Scala (Third Edition). A comprehensive step-by-step guide. https://www.artima.com/shop/programming_in_scala

Ressources en bibliothèque

- Programming in Scala (Third Edition) / Odersky
- Structure and Interpretation of Computer Programs / Abelson

Websites

- https://www.artima.com/shop/programming_in_scala
- https://www.scala-lang.org/
- https://gitlab.epfl.ch/lamp/cs210

PHYS-114

General physics: electromagnetism

Shchutska Lesya

Cursus	Sem.	Type
HES - IN	H	Obl.
HES -SC	H	Obl.
Informatique	BA3	Obl.
Systèmes de communication	BA3	Obl.

Language	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 course first develops the basic laws of electricity and magnetism and illustrates the use in understanding various electromagnetic phenomena.

Content

ELECTRICITY AND MAGNETISM
Electric fields: electric charges and fields; Coulomb's law; Gauss's law
Electric potential and energy: potential; energy; capacitance and capacitors; dielectric materials
Magnetism: magnetic forces and fields; Ampere's law; Biot-Savart law
Electromagnetism: electromotive force; Farady's law; inductance and inductors; Maxwell's equations
Electromagnetic waves: electromagnetic spectrum; antennas

Learning Prerequisites

Recommended courses

General Physics I

Learning Outcomes

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

- Formulate approach for solving physics problems
- Analyze physical systems
- Establish competence in complex problem solving

Transversal skills

- Use a work methodology appropriate to the task.
- Take feedback (critique) and respond in an appropriate manner.
- Access and evaluate appropriate sources of information.

Teaching methods

Ex cathedra with demonstrations, exercises in class

Assessment methods

only final written exam

Supervision

Assistants	Yes

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	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

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

Exercices 30%, examens intermediaire et 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

Sites web

- http://liawww.epfl.ch/
- http://moodle.epfl.ch/

Préparation pour

Intelligent Agents

EPFL

CS-213

Interaction personne-système

Dillenbourg Pierre

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

Langue	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

L'efficacité d'un logiciel pour un tâche ne dépend pas seulement de la performance du logiciel mais aussi de la capacité de l'utilisateur à l'exploiter correctement.Cette capacité résulte du niveau de l'utilisateur mais aussi de la bonne conception de l'interface.

Contenu

1. Introduction à HCI
2. Styles d'##interaction
3. Mécanismes de jeux (1)
4. Introduction à HRI
5. Interactions multimodales
6. Interactions haptiques
7. Design des interfaces
8. Mécanismes de jeux (2)
9. Visualisation des données
10. Anthropomorphisme
11. Accessibilité
12. ##Usability Testingâ#
13. Conception d'##expériences
14. Synthèse

Compétences requises

Concepts importants à maîtriser

Humans use the technologies we build

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

Méthode d'enseignement

Cours ex-cathedra incluant des expériences
Projet : développement d'un jeu multi-utilisateurs avec des robots. Cellulos au moyen de la platforme Unity 3D.

Travail attendu

Développer un jeu impliquant 2 robots haptiques, à partir de la platforme Unity

Méthode d'évaluation

Projet (50%)
Examen écrit (50%)

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Préparation pour

CS-486 Interaction design

COM-308

Internet analytics

Grossglauser Matthias

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

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
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

Stochastic models in communication (COM-300)

Recommended courses

Basic linear algebra
Algorithms & data structures

Important concepts to start the course

Graphs; linear algebra; Markov chains; Java

Learning Outcomes

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

- Explore real-world data from online services
- Develop frameworks 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

Teaching methods

Ex cathedra + homeworks + lab sessions

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 20%, midterm 30%, final exam 50%

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
M.E.J. Newman: Networks: An Introduction, Oxford, 2010

Websites

- http://icawww1.epfl.ch/ix/

EPFL

BIO-109

Introduction aux sciences du vivant (pour IC)

Zufferey Romain

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

Langue	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.
- compositon en bases d'un génome entier, deuxième loi de parité de Chargaff, variations locales de la densité en CpG
- optimisation des codons dans diverses application pratiques.

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 en relation avec les Sciences de la Vie
- 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.

CS-341 **Introduction to computer graphics**

Cursus	Sem.	Type
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	BA6	Opt.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA6	Opt.

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

Remark

Pas donné en 2021-22

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

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

Exercises and Project, Final Examination

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

A list of books will be provided at the beginning of the class

Ressources en bibliothèque

- Polygon mesh processing / Botsch

Notes/Handbook

Slides and online resources will be provided in class

Websites

- http://lgg.epfl.ch/ICG

Prerequisite for

Advanced Computer Graphics

CS-322

Introduction to database systems

Ailamaki Anastasia, Koch Christoph

Cursus	Sem.	Type
Energie et durabilité	MA2, MA4	Opt.
HES - IN	E	Obl.
HES -SC	E	Obl.
Informatique	BA6	Obl.
Mineur en Data science	E	Opt.
Mineur en Informatique	E	Opt.
Sciences et ingénierie de l'environnement	MA2, MA4	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course provides a deep understanding of the concepts behind data management systems. It covers fundamental data management topics such as system architecture, data models, query processing and optimization, database design, storage organization, and transaction management.

Content

This course provides a deep understanding of the concepts behind data management systems.
During this course, the students will learn about:

- The Entity-relationship and Relational Models
- Relational Algebra and Calculus
- The SQL Query Language
- Traditional and Modern Data Storage, File Organizations, and Indexing
- Hashing and Sorting
- Query Evaluation and Relational Operators
- Query Optimization
- Schema Refinement
- Transaction Management (Concurrency Control and Recovery)

Homework
Homeworks will be assigned to aid and assess comprehension of the above material. Homework will be either done using pen and paper or they will be programming exercises. During the semester the students will be asked to do a project to gain experience on how to build a database application, and to apply what they learn in class.

Keywords

databases, database design, data modeling, normalization, database management systems (DBMS), files, indexes, storage, external sorting, queries, query evaluation, query optimization, transactions, concurrency, recovery, SQL

Learning Prerequisites

Required courses

Data structures

Recommended courses

For the practical part of the course (project) the following skills will be needed:

• System oriented programming, with focus on scripting languages to enhance the parsing process of raw data.

• Building user interfaces, either web (e.g., PHP, JSP, ASP, ...) or application GUI (e.g., java).

Important concepts to start the course

Before the beginning of the course students must be familiar with:

• Data structures

• Algorithms concepts

Learning Outcomes

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

• Express application information requirements

• Use a relational DBMS

• Create a database on a relational DBMS

• Design a database with a practical application in mind

• Model the data of an application using ER and relational modeling

• Explore how a DBMS performs its work

• Report performance and possible optimizations for applications using DBMS

• Justify design and implementation choices

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.

• Write a scientific or technical report.

• Make an oral presentation.

Teaching methods

Ex cathedra; including exercises in class, practice with pen and paper or with a computer, and a project

Expected student activities

During the semester, the students are expected to:

• attend the lectures in order to ask questions and interact with the professor,

• attend the exercises session to solve and discuss exercises abou the recently taught material,

• work on a project during the semester which covers the practical side of building an application using a database system,

• take a midterm

• take a final exam

Assessment methods

Homework, project, written examinations and continuous control.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Slides, list of books, additional material (research articles), all indicated and/or available on moodle page.

Ressources en bibliothèque

- Database Management Systems / Ramakrishnan

Notes/Handbook

The slides that are used in the class are available for the students.

Websites

- http://dias.epfl.ch/courses
- http://moodle.epfl.ch/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=198

CS-233(a) **Introduction to machine learning (BA3)**

Salzmann Mathieu

Cursus	Sem.	Type
HES -SC	H	Obl.
Informatique	BA3	Opt.
Sciences et ingénierie de l'environnement	BA5	Opt.
Systèmes de communication	BA3	Opt.

Language	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

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.

Keywords

Machine learning, classification, regression, algorithms

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 (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
- 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-world problems, and rigorously evaluate their performance using

cross-validation.

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.

Teaching methods

- Lectures
- Lab sessions

Expected student activities

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

Assessment methods

- Two graded exercise sessions (10% each).
- Final exam (80%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Course website

EPFL

CS-233(b) **Introduction to machine learning (BA4)**

Fua Pascal

Cursus	Sem.	Type
HES -SC	E	Obl.
Informatique	BA4	Opt.
Sciences et ingénierie de l'environnement	BA6	Opt.
Systèmes de communication	BA4	Opt.

Language	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	

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 : K nearest neighbors, data representation, basic optimization.
- Linear models : Linear regression, least-square classification, logistic regression, linear SVMs.
- Nonlinear method : Polynomial regression, kernel methods.
- Deep learning : Multi-layer pereceptron, CNNs.
- Unsupervised learning : Dimensionality reduction, clustering.

Keywords

Machine learning, classification, regression, algorithms

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 (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
- 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.

Teaching methods

- Lectures
- Lab sessions

Expected student activities

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

Assessment methods

- Continuous control (graded labs)
- Written final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	Course website

EPFL

CS-307

Introduction to multiprocessor architecture

Falsafi Babak

Cursus	Sem.	Type
Informatique	BA5	Obl.
Science et ing. computationelles	MA1, MA3	Opt.
Systèmes de communication	BA5	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly
Number of positions	

Summary

Multiprocessors are a core component in all types of computing infrastructure, from phones to datacenters. This course will build on the prerequisites of processor design and concurrency to introduce the essential technologies required to combine multiple processing elements into a single computer.

Content

- Forms of parallelism
- Parallel programming models
- Cache coherence
- Memory consistency
- Synchronization
- Interconnection networks
- Software efficiency & optimization
- GPU architecture & programming

Keywords

Multiprocessors, multicores, manycores, cache coherence, memory consistency models, memory ordering, manycore cache hierarchies, interconnection networks, synchronization, parallelism, GPU

Learning Prerequisites

Required courses

CS-206 Parallelism and concurrency
CS-208 Computer architecture

Important concepts to start the course

Introductory understanding of computer architecture & organization
Basic C/C++ systems programming

Learning Outcomes

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

- Detect and address inefficiencies in parallel software
- Design and evaluate software for multiple parallel platforms
- Design and evaluate hardware for shared memory
- Compare and contrast hardware design choices in parallel platforms

- Demonstrate and describe the operation of snooping and directory coherence protocols

Teaching methods

Lectures, homework and project

Assessment methods

- Programming Assignements - 30%
- Exercises - 30%
- Final exam - 40%

Supervision

Office hours	Yes
Assistants	Yes

Resources

Websites

- https://parsa.epfl.ch/course-info/cs307/

EPFL

CS-323 **Introduction to operating systems**

Kashyap Sanidhya, Payer Mathias

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

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

Summary

Introduction to basic concepts of operating systems.

Content

The purpose of this course is to discuss the design of operating systems, and operating systems concepts. Topics we will cover include operating system organization, system programming, networked and distributed systems, and storage systems. Most of the time will be spent on multi-process systems (processes, interprocess communication, and synchronization), memory organization(paging), resource allocation and scheduling, file systems, and I/O.
Core topics:

- Function and general structure of an operating system.
- Process management.
- Memory management.
- File systems.
- Virtualization and virtual machines.
- Security aspects of operating systems

Keywords

Operating systems

Learning Prerequisites

Required courses

CS-206 Parallelisme and concurrency
CS-207 Programmation orientée système
CS-212 Projet programmation système

Learning Outcomes

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

- Manage key components of operating systems
- Interpret virtualization of resources
- Discriminate persistence policies

- Manage concurrency between tasks
- Specify security aspects of operating systems

Teaching methods

Lectures and exercises.

Expected student activities

Attendance at lectures and completing exercises.

Assessment methods

The students are assessed both on their theoretical knowledge about operating systems as well as based on implementing parts of an operating system.

- Theoretical assessments during the semester in the forum of weekly quizzes (10% of the grade)
- Practical assessments through several programming labs during the semester (50% of the grade)
- Theoretical assessments in the form of a final exam in the exam session (40% of the grade).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Operating Systems : Three Easy Pieces, R. Arpaci-Dusseau and A. Arpaci-Dusseau (free online book).
Slides available on Moodle.

Ressources en bibliothèque

- Operating Systems : Three Easy Pieces / Arpaci-Dusseau

Références suggérées par la bibliothèque

- Operating Systems Concepts / Silberschatz

CS-358

Making intelligent things

Koch Christoph

Cursus	Sem.	Type
Informatique	BA4, BA6	Opt.
Systèmes de communication	BA4, BA6	Opt.

Language	English
Credits	6
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Project	6 weekly
Number of positions	**50**

Summary

Interdisciplinary project course that aims at teaching students essential skills in prototyping intelligent. Teams of students choose their own challenge 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 Digital System Design

Important concepts to start the course

Basic programming skills.

This course is a project course with a limited capacity for 50 students in the first year. The goal is to ramp up the scalability of the course (as well as to open it up to non-IC undergrad curricula) in later years once a sufficiently large pool of competent teaching assistants has been trained. In case of student interest exceeding capacity, students will be asked to provide a short project proposal in week 1 of the course based on whose quality students will be selected for the course.

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
- Evaluate how to best integrate computational methods and digital fabrication tools to achieve project goals
- Assess own project progress and devise adaptations of the project plan if necessary
- Provide constructive feedback on other groups' projects
- Design a suitable format and material for public presentation of project outcomes

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

COM-300

Modèles stochastiques pour les communications

Thiran Patrick

Cursus	Sem.	Type
HES -SC	H	Obl.
Informatique	BA5	Opt.
Ingénierie des sciences du vivant	MA1, MA3	Opt.
Mineur en Data science	H	Opt.
Mineur en Systèmes de communication	H	Opt.
Systèmes de communication	BA5	Obl.

Langue	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 40% (sous condition d'organisation en présentiel)
- Examen final 60%.

Encadrement

Office hours	Oui
Assistants	Oui
Forum électronique	Oui

Ressources

Bibliographie

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

Ressources en bibliothèque

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

Polycopiés
Polycopié disponible au début du cours et sur la page moodle du cours.

Sites web
- http://moodle.epfl.ch/course/view.php?id=14236

Liens Moodle
- https://moodle.epfl.ch/course/view.php?id=14236

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

CS-328

Numerical methods for visual computing and ML

Jakob Wenzel

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

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

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

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

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)

EPFL

CS-206 Parallelism and concurrency

Kashyap Sanidhya, Odersky Martin

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	1 weekly
Exercises	1 weekly
Project	2 weekly
Number of positions	

Summary

The course introduces parallel programming models, algorithms, and data structures, map-reduce frameworks and their use for data analysis, as well as shared-memory concurrency.

Content

See https://lara.epfl.ch/w/parcon17:top
Parallel programming & execution models
Functional parallelism
Data-level parallelism
Threads and fork/join parallelism
Synchronization
Threads and Shared Memory in Java
Futures
Large-Scale Parallel programming using Apache Spark

Keywords

Parallelism, threads, synchronization, locks, memory models.

Learning Prerequisites

Required courses

- Functional programming (CS-210)
- Algorithms (CS-250)
- Computer Architecture (CS-208)

Recommended courses

System oriented programming (CS-207)

Important concepts to start the course

Functional programming and functional data structures
Algorithms and data structures

Learning Outcomes

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

- Construct parallel software.
- Perform tuning parallel software.

Teaching methods

Ex cathedra, labs, exercices

Assessment methods

With continuous control

Resources

Notes/Handbook

Lecture notes, copies of the slides

EPFL

COM-302 **Principles of digital communications**

Telatar Emre

Cursus	Sem.	Type
HES -SC	E	Obl.
Informatique	BA6	Opt.
Mineur en Systèmes de communication	E	Opt.
Systèmes de communication	BA6	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	2 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

Websites

- http://moodle.epfl.ch

Prerequisite for

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

EPFL

MATH-232 **Probabilities and statistics**

Abbé Emmanuel, Berthier Raphaël Jean

Cursus	Sem.	Type
HES - IN	E	Obl.
HES -SC	E	Obl.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
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

Mid-term and final exams

Resources

Bibliography

Ross, S. (2012) A first course in probability (9th edition). Pearson.
Aussi disponible en traduction française (PPUR): `Initiation aux probabilités'.
A polycopié of the course notes, with the problems etc., will also be available.

Prerequisite for

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

CS-207

Programmation orientée système

Chappelier Jean-Cédric

Cursus	Sem.	Type
HES -SC	E	Obl.
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Obl.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Cours	1 hebdo
Exercices	2 hebdo
Nombre de places	

Résumé

Cours de programmation en langage C se focalisant sur l'utilisation des ressources système, en particulier la gestion de la mémoire (pointeurs).

Contenu

Initiation à la programmation en C : variables, expressions, structures de contrôle, fonctions, entrées-sorties, ...

Approfondissement des spécificités de la programmation système rudimentaire : gestion de la mémoire (pointeurs), des fichiers et autres entrées/sorties.

Les concepts théoriques introduits lors des cours magistraux seront mis en pratique dans le cadre d'exercices sur machine.

Mots-clés

Programmation, langage C, pointeurs, gestion mémoire

Compétences requises

Cours prérequis indicatifs

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

Concepts importants à maîtriser

bases de programmation

Acquis de formation

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

- Modéliser en langage C une situation simple du monde réelle décrite en Français
- Analyser un problème complexe relatifs aux systèems d'information pour le décomposer en sous-problèmes
- Concevoir des algorithmes résolvant des tâches simple à avancées relatives au systèmes informatiques
- Réaliser de façon autonome une application de petite taille au moyen du langage C
- Analyser du code C pour en décrire le résultat ou le corriger s'il est erroné
- Tester l'adéquation du résultat d'un programme par rapport à la tâche demandée

• Transcrire un algorithme en son programme équivalent en C

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.
• Accéder aux sources d'informations appropriées et les évaluer.
• Utiliser une méthodologie de travail appropriée, organiser un/son travail.
• Recevoir du feedback (une critique) et y répondre de manière appropriée.
• Gérer ses priorités.

Méthode d'enseignement

Ex cathedra, travaux pratiques sur ordinateur

Travail attendu

participation au cours et aux exercices ; travail personnel à la maison.

Méthode d'évaluation

1 exercice à rendre (10 %), 1 série pratique notée (40%) et examen final papier (50%)

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Oui

Bibliographie

Notes de cours

Liens Moodle

• http://moodle.epfl.ch/course/view.php?id=6731

Préparation pour

Introduction au bases de données (CS-322) ; Concurrence (CS-206) ; Systèmes d'exploitation (CS-323) ; Genie logiciel (CS-305)

CS-309

Projet de systems-on-chip

Beuchat René

Cursus	Sem.	Type
Informatique	BA6	Opt.
Systèmes de communication	BA6	Opt.

Langue	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	90h
Semaines	14
Heures	**3 hebdo**
Projet	3 hebdo
Nombre de places	

Résumé

L'objectif de ce cours est de s'approprier les connaissances nécessaires pour réaliser du développement "full stack" depuis le hardware jusqu'au software application et s'exécutant sur un système d'exploitation Linux. La base des laboratoires est une plateforme embarquée basée sur un SOC FPGA & ARM.

Contenu

Ce cours-labos va surtout permettre aux étudiants de comprendre les liens entre une application, un operating system (Linux) et le matériel informatique. Une plateform ayant une FPGA et 2 processeurs ARM intégrés est utilisée comme support des labos.
Ce cours est centré sur des laboratoires pratiques à réaliser par les étudiants. Généralement une introduction d'une heure est suivie de 1 à 3 sessions pour réaliser des mini-projets par groupes.
Lors des laboratoires, les travaux seront effectués sur des cartes FPGA-SOC avec processeurs embarqués sous forme softcore (NIOSII) et/ou hardcore (ARM). Des interfaces programmables simples et complexes seront développées en VHDL pour s'interfacer avec des modules externes à contrôler. La méthodologie pas-à-pas sera utilisée pour arriver à la réalisation de systèmes relativement complexes.
Les sujets suivants seront étudiés et au final une application complète sera implémentée :

1. Analyse du système : Multicore ARMs, FPGA, I/Os, et interfaces programmables spécialisées ;
2. Design et simulation d'une interface programmable en VHDL, pour un ADC permettant de lire des joysticks analogiques, et un contrôle de servo-moteurs ;
3. Design et simulation d'un accélérateur spécifique en VHDL pour une caméra thermique ;
4. Test du système spécifique avec développement de logiciel en C avec des outils de cross-debugging (baremetal coding) ;
5. Boot et test d'un système embarqué, baremetal design ;
6. Installation d'un OS: Adaptation et compilation de Linux pour la carte de laboratoire ;
7. Mini-projet final où les étudiants utiliseront leurs connaissances pour capturer une image thermique et à l'aide de joystick définiront divers paramètres et afficheront les résultats sur un affichage LCD ou écran VGA.

Mots-clés

Micro-controllers, Micro-contrôleurs, FPGA, SOC, Embedded Systems, Logic Analyzer, Oscilloscope, projects, work in groups, C, VHDL.

Compétences requises

Cours prérequis obligatoires

- CS-173: Digital system design / Conception de systèmes numériques
- CS-208: Computer architecture
- CS-209: Architecture des Systems-on-Chip

Cours prérequis indicatifs

- CS-307: Introduction to multiprocessor architecture

Concepts importants à maîtriser

Les étudiants doivent connaître l'architecture d'un processeur, d'un système informatique, quelques notions de programmation en C, en langage assembleur et en VHDL.

Acquis de formation

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

- Analyser le problème à résoudre ;
- Planifier le travail à réaliser ;
- Créer l'architecture du système à réaliser ;
- Coordonner le travail ;
- Concevoir le système à réaliser ;
- Implémenter la solution ;
- Intégrer les modules disponibles ;
- Expliquer la solution développée.

Compétences transversales

- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Fixer des objectifs et concevoir un plan d'action pour les atteindre.
- Communiquer efficacement et être compris y compris par des personnes de languages et cultures différentes.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Faire preuve d'esprit critique
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.

Méthode d'enseignement

- Travail par groupes d'étudiants ;
- Laboratoires pratiques ;
- Mini-projets spécifiques par groupe ;
- Travaux avec des outils de développement croisé et de systèmes matériels réels ;
- Utilisation des outils de débogage tels que oscilloscopes et analyseurs logiques ;
- Les travaux sont réalisés sur les systèmes matériels réels avec des micro-contrôleurs et FPGA.

Travail attendu

- Brainstorming par groupe ;
- Répartition de la charge de travail à travers le groupe ;
- Gestion de la planification ;
- Analyse des données des data-sheet des composants utilisés ;
- Résolution de problèmes pratiques sur des systèmes matériels ;
- Développement de modules FPGA en VHDL ;
- Développement de logiciels en C et observation en assembleur du déroulement du programme ;

• Configuration et construction d'un Linux sur un système embarqué.

Méthode d'évaluation

1. Rapports réguliers (60%)
2. Présentation orale finale (20%)
3. Démonstration (15%)
4. Tenue d'un journal de laboratoire (5%)

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Non

Sites web

• http://wiki.epfl.ch/prsoc

Liens Moodle

• http://moodle.epfl.ch/course/view.php?id=14480

Préparation pour

CS-473: Embedded Systems
CS-476: Real Time Embedded systems

EPFL

COM-307

Projet en systèmes de communication I

Profs divers *

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

Langue	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'un professeur ou d'un assistant.

Contenu

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

https://www.epfl.ch/schools/ic/fr/education-fr/bachelor-fr/systemes-communication/projets-labo-sc/

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 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

Service de cours virtuels (VDI)

Non

Sites web

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

EPFL

CS-212 Projet programmation système

Bugnion Edouard, Chappelier Jean-Cédric

Cursus	Sem.	Type
Informatique	BA4	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA4	Opt.

Langue	français
Crédits	2
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	60h
Semaines	14
Heures	**2 hebdo**
Projet	2 hebdo
Nombre de places	

Résumé

L'objectif de ce cours à projet est de donner aux étudiants une expérience de la pratique de la programmation système : écriture, correction, amélioration et analyse critique de leur code.

Contenu

- Ce cours sera enseigné en parallèle du cours « Programmation Orientée Système » (CS-207). Il offre aux étudiants l'opportunité de développer dans un cadre pratique à large échelle les concepts présentés dans cet autre cours. Les étudiants devront en effet non seulement développer leur propre code à partir de rien sur un cas concret, mais aussi lire du code professionnel développé par d'autres de sorte à pouvoir s'en inspirer.
- Ce cours consistera en un projet constitué de plusieurs parties réparties sur le semestre. Il insistera sur les concepts clés présentés dans les autres cours liés au domaine de la programmation système (systèmes de fichiers, réseaux, accès mémoire, concurrence, …), mais mettra aussi en place de façon pratique plusieurs élémens qui seront approfondis plus tard dans le cursus (aspects réseaux avancés, planification, etc.) dans le but d'offrir aux étudiants une première approche pratique à ces concepts.

Mots-clés

- programmation système, gestion mémoire, système de fichiers

Compétences requises

Cours prérequis obligatoires

- Programmation Orientée Système (CS-207)
- Parallelism & Concurrency (CS-206)

Cours prérequis indicatifs

- Computer Networks

Concepts importants à maîtriser

- programmation
- parallélisme et concurrence

Acquis de formation

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

- Modéliser en langage C des problèmes système typiques
- Analyser des problèmes complexes relatifs aux sytèmes informatiques et les décomposer en sous-problèmes
- Réaliser des applications de taille moyenne en langage C
- Analyser des projet en langage C pour être capable de comprendre ce qu'ils font et comment ils sont organisés
- Tester les résultat d'un projet en langage C et le corriger là où cela est nécessaire

Compétences transversales

- Accéder aux sources d'informations appropriées et les évaluer.
- Recevoir du feedback (une critique) et y répondre de manière appropriée.
- Utiliser les outils informatiques courants ainsi que ceux spécifiques à leur discipline.
- Gérer ses priorités.
- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.

Méthode d'enseignement

projet

Travail attendu

- écrire le code et la documentation d'un projet de groupe

Méthode d'évaluation

- rendu du projet en plusieurs étapes progressives au cours du semestre.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Préparation pour

- CS-323 (Operating Systems)

COM-303

Signal processing for communications

Prandoni Paolo

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
HES -SC	E	Obl.
Informatique	BA6	Opt.
Mineur en Systèmes de communication	E	Opt.
Science et ing. computationelles	MA2, MA4	Opt.
Systèmes de communication	BA6	Obl.

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

Summary

Students learn digital signal processing theory, including discrete time, Fourier analysis, filter design, adaptive filtering, sampling, interpolation and quantization; they are introduced to image processing and data communication system design.

Content

1. Basic discrete-time signals and systems: signal classes and operations on discrete-time signals, signals as vectors in Hilbert space
2. Fourier Analysis: properties of Fourier transforms, DFT, DTFT; FFT.
3. Discrete-Time Systems: LTI filters, convolution and modulation; difference equations; FIR vs IIR, stability issues.
4. Z-transform: properties and regions of convergence, applications to linear systems.
5. Filter Design: FIR design methods, IIR design methods, filter structures.
6. Stochastic and Adaptive Signal Processing: random processes, spectral representation, Optimal Least Squares adaptive filters.
7. Interpolation and Sampling: the continuous-time paradigm, interpolationthe sampling theorem, aliasing.
8. Quantization: A/D and D/A converters.
9. Multi-rate signal processing: upsampling and downsampling, oversampling.
10. Multi-dimensional signals and processing: introduction to Image Processing.
11. Practical applications: digital communication system design, ADSL.

Keywords

signal processing, discrete-time, continuous-time, filter, filter design, sampling, aliasing, DSP, Fourier transform, FFT, modem, ADSL

Learning Prerequisites

Required courses

calculus, linear algebra

Recommended courses

Circuits and systems, basic probability theory

Important concepts to start the course

vectors and vector spaces, functions and sequences, infinite series

Learning Outcomes

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

- Identify signals and signal types
- Recognize signal processing problems
- Apply the correct analysis tools to specific signals
- Check system stability
- Manipulate rational transfer functions
- Implement signal processing algorithms
- Design digital filters
- Interpret complex signal processing systems

Transversal skills

- Use a work methodology appropriate to the task.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Use both general and domain specific IT resources and tools

Teaching methods

Course with exercises sessions and coding examples and exercises in Python (Jupyter Notebooks)

Expected student activities

complete weekly homework, explore and modify Jupyter Notebook examples

Assessment methods

final exam fully determines final grade.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Signal processing for Communications, EPFL Press, 2008, by P. Prandoni and M. Vetterli. The book is available for sale in printed form online and in bookstores; in iBook format on the Apple store and is also available as a free pdf file at http://www.sp4comm.org/

Ressources en bibliothèque

- Signal processing for Communications / Prandoni

Références suggérées par la bibliothèque

- signal rpocessing for communications

Notes/Handbook

lecture slides available for download at the beginning of the semester.
A complete online DSP MOOC is available on Coursera.

Websites

- http://www.sp4comm.org/
- https://www.coursera.org/learn/dsp1/

- https://www.coursera.org/learn/dsp2/
- https://www.coursera.org/learn/dsp3/
- https://www.coursera.org/learn/dsp4/

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15139

Prerequisite for

adaptive signal processing, image processing, audio processing, advanced signal processing

EE-205

Signals and systems (for EL&IC)

Shkel Yanina

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

Language	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	

Summary

This class teaches the theory of linear time-invariant (LTI) systems. These systems serve both as models of physical reality (such as the wireless channel) and as engineered systems (such as electrical circuits, filters and control strategies).

Content

The design of advanced systems (such as WiFi, cell phones, drones, airplanes) requires a thorough theoretical underpinning. This class teaches one of the most powerful and important pillars: The theory of linear time-invariant (LTI) systems. These systems serve both as models of physical reality (such as the wireless channel) and as engineered systems (such as filters and control strategies).
The class will cover the following topics :

1. Systems: Definitions (1 week)
2. LTI Systems (3 weeks)
3. The Frequency Response of stable LTI Systems (1 week)
4. Fourier Techniques for stable LTI Systems (3 weeks); with applications to Communication Systems and Signal Processing
5. Laplace and Z-Transform Techniques for LTI Systems (5 weeks); with applications to Control Systems

Keywords

Systems, Circuits, Signals, Frequency Response, Transfer Function, Fourier Transform, Laplace Transform, Z Transform, Stability, Causality, Sampling

Learning Prerequisites

Required courses

Analysis I, II, III. Linear algebra I.

Recommended courses

Linear algebra II

Learning Outcomes

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

- Describe properties of LTI systems
- Solve for poles and zeros of LTI systems
- Recall properties of CT Fourier transform
- Analyze LTI systems by spectral analysis

- Operate with Fourier transform tools
- Work out / Determine impulse response of CT LTI

Teaching methods

- Classroom lectures
- Written exercises
- Graded homework problems

Expected student activities

- Read course book in english (the course is taught in english)

Assessment methods

Homeworks and written mid-term exam and final exams

Resources

Bibliography

The following is a recommended (but not required) book:
A. V. Oppenheim and A. S. Willsky, with S. Hamid Nawab, Signals and Systems. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1996.

Ressources en bibliothèque

- A. V. Oppenheim and A. S. Willsky, with S. Hamid Nawab, Signals and Systems

Notes/Handbook

will be made available

CS-306

Software development project

Candea George

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA6	Obl.
Mineur en Informatique	E	Opt.
Systèmes de communication	BA6	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Project	4 weekly
Number of positions	

Summary

This course teaches the basics of developing real-world software, i.e., software that is large and complex, is developed by a team, evolves and needs to be maintained, and can cause serious harm if it fails. Students develop an Android app and do a lot of programming.

Content

- Problem solving for software engineers
- Development processes, in particular agile methods
- Tools (source control, project management, issue trackers, debuggers, etc.)
- Android

Learning Prerequisites

Required courses

- CS-305 Software Engineering (strict requirement, no exceptions)

Important concepts to start the course

First and foremost, students taking Software Engineering must be proficient Java programmers. Without a good prior knowledge of Java, it is very difficult to keep up with the pace of the class. Familiarity with Android development is a plus but not required.

Learning Outcomes

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

- Design mobile apps
- Coordinate a team of developers

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Assess progress against the plan, and adapt the plan as appropriate.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

- Manage priorities.
- Take feedback (critique) and respond in an appropriate manner.

Teaching methods

Team-based project

Expected student activities

Work with team members to complete a substantial project

Assessment methods

Throughout the semester (contrôle continu).
Grade determined based on both team and individual performance in the project.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

- http://sweng.epfl.ch/

CS-305

Software engineering

Candea George

Cursus	Sem.	Type
HES - IN	H	Obl.
Informatique	BA5	Obl.
Mineur en Informatique	H	Opt.
Science et ing. computationelles	MA1, MA3	Opt.
Systèmes de communication	BA5	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	1 weekly
Project	1 weekly
Number of positions	

Summary

This course teaches the basics of modern software development: designing software, working in a team, writing good code, shipping software, and evolving software. It emphasizes building software that meets high standards of quality, reliability, security, and manageability.

Content

Writing software

- Modularity
- Interfaces
- Software architecture

Getting software right

- Requirements
- Testing
- Verification
- Debugging
- Security
- Performance

Shipping software

- Development processes
- DevOps
- Software evolution

Continuous and independent learning is essential to being a good software engineer because, unlike mathematics or physics, the field changes fast. This course prepares students to become lifelong auto-didacts who build upon the foundation of immutable principles that govern good software engineering.

Keywords

design patterns, fault tolerance, software testing, code analysis, software verification, security, performance, usability, refactoring, agile development methods, version control systems, continuous integration

Learning Prerequisites

Required courses

- CS-108 Practice of Object-Oriented Programming
- CS-206 Parallelism and Concurrency
- CS-207 System-oriented Programming
- COM-208 Computer Networks
- CS-208/209 Computer Architecture
- CS-210 Functional Programming

Students who do not master the material taught in the prerequisite courses prior to starting CS-305 typically do not manage to pass this course.

Important concepts to start the course

Students are required to have good programming skills in an object-oriented language (e.g., Java).

Learning Outcomes

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

- Design software that is reliable, secure, user-friendly, and performs well
- Implement sophisticated designs and algorithms
- Specify requirements for software systems
- Develop code that is maintainable
- Organize a team to execute a medium-sized software project
- Assess / Evaluate design and implementation alternatives

Teaching methods

- Combination of online and in-class lectures
- Online textbook
- Homework exercises

Expected student activities

- Attend and actively participate in lectures
- Read and understand assigned materials
- Complete homework exercises independently

Assessment methods

- 20% based on online quizzes and homeworks (during the semester)
- 80% based on a final exam (during the exam session)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography
Please see the course website for the latest information and up-to-date bibliography

Ressources en bibliothèque
-
-

Websites
- https://sweng.epfl.ch

EPFL

CS-234

Technologies for democratic society

Ford Bryan Alexander

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

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	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 exericses, 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://moodle.epfl.ch/course/view.php?id=15738

CS-251

Theory of computation

Göös Mika

Cursus	Sem.	Type
HES - IN	E	Obl.
Informatique	BA4	Obl.
Systèmes de communication	BA4	Obl.

Language	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	

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

COM-309 Traitement quantique de l'information

Macris Nicolas

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

Langue	français
Crédits	4
Session	Hiver
Semestre	Automne
Examen	Ecrit
Charge	120h
Semaines	14
Heures	**4 hebdo**
Cours	3 hebdo
Exercices	1 hebdo
Nombre de places	

Résumé

L'information est traitée et stockée dans des composants matériels. Avec leur miniaturisation, il faut remplacer le concept de bit classique par celui de bit quantique. Ce cours développe le sujet des communications, de la cryptographie et des corrélations quantiques. La machine IBM Q sera abordée.

Contenu

Introduction a la mecanique quantique des systemes discrets.
- Polarisation des photons, états quantiques, règle de Born.
- Expérience de Stern-Gerlach, spin 1/2, états quantiques sur la sphère de Bloch.
- Dynamique du spin, Oscillations de Rabi et manipulations de l'état quantique.
- Notion abstraite de qubit. Etats à plusieurs qubits.

Cryptographie, Communications et Corrélations
- Génération d'une clé secrète: protocoles BB84 et B92.
- Intrication: paires de Einstein-Podolsky-Rosen.
- Inégalités de Bell. Expériences d'Aspect-Grangier. Protocole de Ekert pour une clé secrète.
- Protocoles de téléportation et dense coding.

Mots-clés

Polarisation, spin, bit quantique, intrication, téléportation, cryptographie quantique.

Compétences requises

Cours prérequis obligatoires

Algèbre linéaire.

Cours prérequis indicatifs

physique de base: mécanique et ondes.

Concepts importants à maîtriser

Matrices, valeurs et vecteurs propres, produit scalaire, manipulations algébriques de base avec des nombres complexes.

Acquis de formation

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

- Expliquer les principes de la physique quantique pour les systèmes discrets
- Expliquer le concept de qubit et donner quelques exemples

- Décrire comment manipuler des qubits
- Connaitre les protocoles de base de la cryptographie quantique.
- Connaitre les protocoles de dense coding et de téléportation.
- Expliquer ce qu'est l'intrication.
- Etre familier avec les ordinateurs IBM Q

Méthode d'enseignement

Ex-Cathedra. Séances d'exercices. Discussions des lectures pédagogiques proposées aux étudiants.

Travail attendu

Participation au cours. Résolution d'exercices. Lectures de revues pédagogiques.

Méthode d'évaluation

miniprojet on IBM Q machine, graded homeworks, examen final écrit.

Ressources

Bibliographie

David Mermin, ***Quantum computer science, An introduction*****,** *Cambridge university press 2000.* Livre écrit pour des informaticiens et qui ne requiert pas de connaissances en physique..
Michel Le Bellac, ***A short introduction to quantum information and quantum computation,*** *Cambridge University Press. Pour l'édition en francais voir Editions Belin 2005.* Un livre pédagogique qui introduit quelques aspects physiques elementaires du sujet.
Neil Gershenfeld. ***The physics of information technology.*** *Cambridge University Press*. Un livre original sur les technologies de base utiles en informatique et communication classique et/ou quantique.

Ressources en bibliothèque

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

Polycopiés

Notes fournies en classe. Revues sur le sujet. exercices et corrigés fournis en cours d'année.

Sites web

- https://ipg.epfl.ch/doku.php?id=en:courses

Préparation pour

Calcul quantique

CS-450

Advanced algorithms

Kapralov Mikhail

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Obl.
Data Science	MA2, MA4	Obl.
Informatique et communications		Opt.
Informatique	MA2, MA4	Obl.
Mineur en Data science	E	Opt.
Mineur en Informatique	E	Opt.
Robotique, contrôle et systèmes intelligents		Opt.
SC master EPFL	MA2, MA4	Obl.
Science et ing. computationelles	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
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.

Keywords

See content.

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

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes
Others	For details, see the course web page.

Resources

Bibliography

See web page for the course.

Ressources en bibliothèque

- Randomized Algorithms / Motwani
- Approximation Algorithms / Vazirani
- Quantum Computation and Quantum Information / Nielsen
- Algebraic Complexity Theory / Buergisser
- Computational Complexity / Papadimitrou

Notes/Handbook

Class notes and references for the running semester will be provided as needed within a few days after each lecture.

Websites

- http://theory.epfl.ch/courses/AdvAlg/

EPFL

CS-470

Advanced computer architecture

Ienne Paolo

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Obl.
Génie électrique et électronique	MA2, MA4	Opt.
Informatique	MA2, MA4	Obl.
Mineur en Informatique	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Project	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 Architecture des ordinateurs or Computer Architecture I

Recommended courses

• CS-209 Architecture des systèmes-on-chip or 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

Labs, homeworks, and final exam.

Supervision

Office hours	No
Assistants	Yes
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://moodle.epfl.ch/course/view.php?id=15017

Prerequisite for

- CS-471 Advanced Multiprocessor Architecture

CS-440

Advanced computer graphics

Jakob Wenzel

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Humanités digitales	MA2, MA4	Opt.
Informatique	MA2, MA4	Opt.
Mineur en Informatique	E	Opt.
Mineur en Systèmes de communication	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

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 (60%), final project (40%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

A list of books will be provided at the beginning of the class

Ressources en bibliothèque

- Physically Based Rendering: From Theory to Implementation / Pharr

Notes/Handbook

Slides and online resources will be provided in class

Websites

- https://rgl.epfl.ch/courses/ACG22

COM-501

Advanced cryptography

Vaudenay Serge

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Mineur en Data science	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

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
- A computational introduction to number theory and algebra / Shoup
- Communication security / Vaudenay

Websites

- http://lasec.epfl.ch/teaching.shtml

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=13913

EPFL

CS-471 **Advanced multiprocessor architecture**

Falsafi Babak

Cursus	Sem.	Type
Cybersecurity	MA1, MA3	Opt.
Génie électrique et électronique	MA1, MA3	Opt.
Informatique et communications		Obl.
Informatique	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
Science et ing. computationelles	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	4 weekly
Number of positions	

Remark

Cours biennal donné une année sur deux les années impaires

Summary

Multiprocessors are now the defacto building blocks for all computer systems. This course will build upon the basic concepts offered in Computer Architecture I to cover the architecture and organization of modern multiprocessors from mobile and embedded platforms to servers, data centers and cloud computing platforms.

Content

Introduction to multiprocessor systems, parallel programming models including Pthreads, MPI, hardware and software transactional memory, synchronization primitives, memory consistency mdels, cache coherence, on-chip shared cache architectures, on-chip interconnects, multi-chip interconnects, multi-chip bus-based and general-purpose interconnect-based shared-memory systems, clusters.
The course will include weekly readings, discussions, and student reviews and reports on publications (besides the text book) of seminal and recent contributions to the field of computer architecture. Student reviews, class discussions, and an independent research project will account for a significant fraction of the grade. Feedback on performance will be given only upon request by a student. There will be no recitation classes.
The course will also include an independent and original research project, in which students study, improve, and evaluate multiprocessor innovations using a software simulation infrastructure. There will be a list of project ideas given out, but students can suggest and work on their own ideas with potentials for advancing the state of the art.

Learning Prerequisites

Recommended courses

Computer Architecture I, basic C/C++ systems programming.

Learning Outcomes

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

- Design and evaluate parallel computer organizations
- Develop parallel programs and benchmarks for parallel systems
- Design the basic components of modern parallel systems including multiple processors, cache hierarchies and networks
- Quantify performance metrics for parallel systems
- Interpret and critique research papers
- Plan , propose and conduct a research project empirically

• Present research contributions

Teaching methods

Lectures, homeworks, and a research project

Assessment methods

Continuous control :
Homework : 30 %, Project 15 %, Midterm test : 20 %,
End term test : 35 %

Resources

Websites

• https://parsa.epfl.ch/course-info/cs471/

EPFL

COM-417 **Advanced probability and applications**

Lévêque Olivier

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Génie électrique		Opt.
Informatique et communications		Obl.
Informatique	MA2, MA4	Opt.
Mineur en Data science	E	Opt.
Robotique, contrôle et systèmes intelligents		Opt.
SC master EPFL	MA2, MA4	Obl.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 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 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

- 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
- Probability and Random Processes

Notes/Handbook

available on the course website

Websites

- https://moodle.epfl.ch/course/view.php?id=14557

Prerequisite for

Advanced classes requiring a good knowledge of probability

CS-523 Advanced topics on privacy enhancing technologies

Troncoso Carmela

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Informatique et communications		Obl.
Informatique	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
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 protect privacy, and how to evaluate the protection they provide.

Content

The course will cover the following topics:

- Privacy definitions and concepts.
- Privacy-preserving cryptographic 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-402 Information Security and Privacy
COM-301 Computer Security

Recommended courses

COM-401 Cryptography

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 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

Assessment methods

Final exam

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

MATH-493

Applied biostatistics

Goldstein Darlene

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Génie civil & environnement		Opt.
Informatique	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Ingénierie des sciences du vivant	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
Mineur en Data science	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

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 proactical 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.
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.

EPFL

CS-401

Applied data analysis

West Robert

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational Neurosciences minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
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.
Financial engineering	MA1, MA3	Opt.
Internet of Things minor	H	Opt.
Learning Sciences		Obl.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
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, more data vs. advanced algorithms, curse of dimensionality, etc.)
- Text mining: vector space model, topic models, word embedding
- Social network analysis (influencers, community detection, 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 and the in-class quizzes.
In parallel, the students will embark on a semester-long project, split in agile teams of 3-4 students. The project consists of two parts: (1) replication of a data analysis pipeline from a published scientific paper, (2) a "free-style" component where students propose and execute their own extension of part 1. 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 also 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.

Recommended courses

• CS-423 Distributed Information Systems
• CS-433 Machine Learning

Important concepts to start the course

Algorithms, (object-oriented) programming, basic probability and statistics

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, correlation vs. causality, knowledge extraction, critical thinking, team discussions, ad-hoc visualizations, etc.)
• Perform result dissemination (reporting, visualizations, publishing reproducible results, ethical concerns, etc.)

Transversal skills

- Give feedback (critique) in an appropriate fashion.
- Demonstrate the capacity for critical thinking
- 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
- In-class quizzes
- Course project

Expected student activities

Students are expected to:

- Attend the lectures and lab sessions
- Complete 2-3 homework assignments
- Complete 3 in-class quizzes (held during lab sessions)
- Conduct the class project
- Read/watch the pertinent material before a lecture
- Engage during the class, and present their results in front of the other colleagues

Assessment methods

- 30% continuous assessment during the semester (homework)
- 30% final exam, data analysis task on a computer (3 hours)
- 25% final project, done in groups of 4
- 15% regular online quizzes

Supervision

Others	http://ada.epfl.ch

EPFL

CS-456

Artificial neural networks

Gerstner Wulfram

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Electrical Engineering		Opt.
Financial engineering	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

Since 2010 approaches in deep learning have revolutionized fields as diverse as computer vision, machine learning, or artificial intelligence. This course gives a systematic introduction into the main models of deep artificial neural networks: Supervised Learning and Reinforcement Learning.

Content

- *General Introduction: Deep Networks versus Simple perceptrons*

- *Reinforcement Learning 1: Bellman equation and SARSA*
- *Reinforcement Learning 2: variants of SARSA, Q-learning, n-step-TD learning*
- *Reinforcement Learning 3: Policy gradient*

- *Deep Networks 1: BackProp and Multilayer Perceptrons*
- *Deep Networks 2: Regularization and Tricks of the Trade in deep learning*
- *Deep Networks 3: Error landscape and optimization methods for deep networks*
- *Deep Networks 4: Statistical Classification by deep networks*
- *Deep Networks 5: Convolutional networks*

- *Deep reinforcement learning 1: Exploration*
- *Deep reinforcement learning 2: Actor-Critic networks*
- *Deep reinforcement learning 3: Atari games and robotics*
- *Deep reinforcement learning 4: Board games and planning*
- *Deep reinforcement learning 5: Sequences, recurrent networks, partial observability*

Keywords

Deep learning, artificial neural networks, reinforcement learning, TD learning, SARSA,

Learning Prerequisites

Required courses

CS 433 Machine Learning (or equivalent)
Calculus, Linear Algebra (at the level equivalent to first 2 years of EPFL in STI or IC, such as Computer Science, Physics or Electrical Engineering)

Recommended courses

stochastic processes
optimization

Important concepts to start the course

- *Regularization in machine learning,*
- *Training base versus Test base, cross validation.*
- *Gradient descent. Stochastic gradient descent.*
- *Expectation, Poisson Process, Bernoulli Process.*

Learning Outcomes

- Apply learning in deep networks to real data
- Assess / Evaluate performance of learning algorithms
- Elaborate relations between different mathematical concepts of learning
- Judge limitations of algorithms
- Propose algorithms and models for learning in deep networks
- Apply Reinforcement Learning

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

ex cathedra lectures and miniproject. Every week the ex cathedra lectures are interrupted for at least one in-class exercise which is then discussed in classroom before the lecture continues. Additional exercises are given as homework or can be disussed in the second exercise hour.

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

Assessment methods

written exam (70 percent) and miniproject (30 percent)

Supervision

Office hours	Yes

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. Choice between two different exercise sessions

Resources

Bibliography

- Textbook: Deep Learning by Goodfellow, Bengio, Courville (MIT Press)
- Textbook: Reinforcement Learning by Sutton and Barto (MIT Press)

Pdfs of the preprint version for both books are availble online

Ressources en bibliothèque

- Reinforcement Learning by Sutton and Barto
- Deep Learning by Goodfellow, Bengio, Courville

Websites

- http://for videos and lecture slides https://lcnwww.epfl.ch/gerstner/VideoLecturesANN-Gerstner.html
- http://main web page is moodle

Videos

- http://yes, for most session.

EPFL

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	English
Credits	3
Session	Winter
Semester	Fall
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 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).

Content

1. Introduction: Speech processing tasks, language engineering applications.

2. Basic Tools: Analysis and spectral properties of the speech signal, linear prediction algorithms, statistical pattern recognition, dynamic programming.

3. Speech Coding: Human hearing properties, quantization theory, speech coding in the temporal and frequency domains.

4. Speech Synthesis: Morpho-syntactic analysis, phonetic transcription, prosody, speech synthesis models.

5. Automatic Speech Recognition: Temporal pattern matching and Dynamic Time Warping (DTW) algorithms, speech recognition systems based on Hidden Markov Models (HMMs).

6. Speaker recognition and speaker verification: Formalism, hypothesis testing, HMM based speaker verification.

7. Linguistic Engineering: state-of-the-art and typical applications

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.

Learning Outcomes

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

- speech signal properties
- Exploit those properties to speech codign, speech synthesis, and speech recognition

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

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Ressources en bibliothèque

- Traitement de la parole / Boite

Websites

- http://lectures.idiap.ch/

EPFL

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		Obl.
Life Sciences Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	**30**

Summary

The course provides a comprehensive overview of methods, algorithms, and computer tools used in in computational bioimaging and bioimage analysis. It exposes the fundamental concepts and the practical computer solutions to extract quantitative information from multidimensional images.

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 (mainly in ImageJ/Fiji). 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, reconstruction, deconvolution, denoising, stitching, visual feature detection, segmentation, active contours, image analysis workflow, pixel classification, machine learning, and tracking for building a cell lineage.
The course is composed of lectures, workshops with the state-of-the-art software packages, computer sessions (programmation) and a mini-project. A personal laptop is recommended to run (open-source) bioimage software packages.

Keywords

Bioimage, microscopy, image processing, image reconstruction, image analysis, visualization, multidimensional data analysis, learning

Learning Prerequisites

Important concepts to start the course

- Basic knowledge in signal or image processing
- Baisc knowledge in programming

Learning Outcomes

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

- 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

Teaching methods

Lecturing with demonstration, workshops, computer laboratories, hands-on

Assessment methods

Continuous: mid-term and end-term exams and a mini-project

EPFL

BIO-465 **Biological modeling of neural networks**

Gerstner Wulfram

Cursus	Sem.	Type
Auditeurs en ligne	E	Opt.
Biocomputing minor	E	Opt.
Biomedical technologies minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	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.
Neuroprosthetics minor	E	Opt.
Neuroscience		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

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 determinstic or stochatic differnetial equations.

Content

I. Models of single neurons 1. Introduction: brain vs computer 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./6. Associative Memory and Attractor Dynamics (Hopfield Model) 7. Neuronal Populations and networks 8. Continuum models and perception 9. Competition and models of Decision making ***III. Noise and the neural code*** 10. Noise and variability of spike trains (point processes, renewal process, interval distribution) 11: Variance of membrane potentials and Spike Response Models **IV. Plasticity and Learning** 12. Synaptic Plasticity and Long-term potentiation and Learning (Hebb rule, mathematical formulation) 13. Summary: Fitting Neural Models to Data

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, stochastic processes,

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
- Formulate stochastic models of biological phenomena
- Formalize biological facts into mathematical models
- Prove stability and convergence
- Apply model concepts in simulations
- Predict outcome of dynamics
- Describe neuronal phenomena

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Collect data.
- Write a scientific or technical report.

Teaching methods

Classroom teaching, exercises and miniproject. One of the two exercise hours is integrated into the lectures.

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%)

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 / Gerstner

Notes/Handbook

The textbook is online at: http://neuronaldynamics.epfl.ch/

Videos

- http://lcn.epfl.ch/~gerstner/NeuronalDynamics-MOOC1.html
- http://lcn.epfl.ch/~gerstner/NeuronalDynamics-MOOC2.html

EPFL

EE-512

Biomedical signal processing

Vesin Jean-Marc

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.
SC master EPFL	MA1, MA3	Opt.

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

Summary

The goal of this course is to introduce the techniques most commonly used for the analysis of biomedical signals, and to present concrete examples of their application for diagnosis purposes.

Content

1. Generalities on biomedical signal processing
2. Digital signal processing - basics

- sampling
- Fourier transform
- filtering
- stochastic signals correlation, and pwoer spectral density

3. Time-frequency analysis

- short-term Fourier transform
- time-frequency distributions, Cohen's class
- wavelet transform

4. Linear modeling

- autoregressive models
- linear prediction
- parametric spectral estimation
- criteria for model selection

5. Adaptive filtering

- adaptive prediction
- adaptive estimation of transfert functions
- adaptive interference cancellation

6. Miscellaneous

- polynomial models
- singular value decomposition
- principal component analysis

Keywords

signal processing, biomedical engineering, signal modeling, spectral analysis, adaptive filtering

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

Teaching methods

lectures, lab sessions using Matlab

Assessment methods

1 point for lab/exercise sessions reports
2 exams: end of November 2points - final exam 3 points

Supervision

Office hours	Yes
Assistants	Yes

CS-490

Business design for IT services

Wegmann Alain

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	3 weekly
Number of positions	

Summary

In this course, students practice designing digital business services. The students learn to apply the principles of technical sales while developing their own projects. Through this process, students gain insight into the co-creation of relationships between a company and its customers.

Content

This course is for engineers who are interested in creating a better relationship with the users of the engineers'technology. The experience the students gain is useful especially in the context of startups. This is also relevant for intrapreneur (engineers who seek to bring innovation in companies) and for anyone who seeks to establish and evolve their business relationships.

The course participants begin their projects by *exploring* an area that interests them. They *evaluate* with whom they can establish a company/customer relationship. They *engage* in a concrete relationship with someone interested in their idea. They *establish* the relationship by developing, delivering, and measuring the service. And they *evolve* these relationships through the digitalization of these services. These are not necessarily chronological stages of journey that the engineer needs to progress through.They are the five *experiences* that the students practice in the course.

To complete this journey, we provide the tools to help the course participants to develop the four following capabilities: (i) the *curiosity* - the receptivity to uncovering the unexpected, (ii) the *awareness* - the capacity to assess a reality in order to optimize solutions (iii) the *service understanding* - the capacity to represent what the company exchanges with its customers, (iv) *the actions* planning - *the* capacity *to* make the result concrete through *actions*.

Indeed, this is the SEAM method. We began developing it 1997, at EPFL. SEAM is a system thinking method that reconciles multiple theories from various disciplines from business engineering. SEAM is an open method.

Come *practice* with us! It might be uncomfortable and difficult, yet fun and interesting. By transcending this, you will learn new tools to reframe problems that are apparently impossible to solve.

Keywords

Entrepreneurial motivation, information inquiry, ecosystem analysis, competitor analysis.

Business definition, business services, IT services, segmentation, lead, prospect, opportunity, qualification, value proposition, quotation, break-even model.

The SEAM canvas, customer-relation management tool.

Learning Outcomes

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

- Create a precise and detailed description for a new business design
- Analyze environmental as well as organizational factors in a business design
- Design a business model in details (service, value, finance)
- Assess / Evaluate alternative business strategies
- Synthesize multiple marketing theories (from seminal publications)
- Represent the key concepts of a business design (ecosystem, value, finance)
- Interpret evidences collected through extensive interviews
- Investigate innovative views of a business design

Transversal skills

- Collect data.
- Access and evaluate appropriate sources of information.
- Write a scientific or technical report.
- Make an oral presentation.
- Summarize an article or a technical report.

Teaching methods

Experiential learning + individual work

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.
Neuroprosthetics minor	H	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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

Ressources en bibliothèque

- Essential Cell Biology / Alberts

EPFL

CS-524 **Computational complexity**

Göös Mika

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 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=NL)
- Boolean circuits and nonuniform computation
- Power of randomness (interactive proofs, IP=PSPACE)
- 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

- 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.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

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 / Stasys

Websites

- http://theory.epfl.ch/courses/complexity/

EPFL

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.
Electrical and Electronical Engineering	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	During the semester
Workload	150h
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

- Introduction to Computer Vision.
- Signal Processing for Communications.
- Machine Learning.

Important concepts to start the course

- Basic signal/image processing.
- Basic computer vision.
- Basic programming (Python, iOS, Android).

Learning Outcomes

- 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.
- 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.

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.

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

- Selected book chapters
- Course notes (on moodle)
- Links to relevant scientific articles and on-line resources will be given on moodle.

CS-442

Computer vision

Fua Pascal

Cursus	Sem.	Type
Communication systems minor	E	Opt.
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.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
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 matlab.

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

• Computer Vision: Algorithms and Applications / Szeliski
• Multiple View Geometry in Computer Vision / Zisserman

Websites

• http://cvlab.epfl.ch/

Moodle Link

• http://moodle.epfl.ch/course/view.php?id=472

EPFL

COM-418

Computers and music

Prandoni Paolo

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

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

Required courses

digital signal processing, programming

Recommended courses

signals and systems, Python, C++

Important concepts to start the course

Digital signals, filters, spectral analysis

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

Bibliography

TBD

Notes/Handbook

handouts, papers and code samples

CS-453

Concurrent algorithms

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

Summary

With the advent of multiprocessors, 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
A multicore architect
Processes and objects
Safety and liveliness

Parallel programming
Automatic parallelism
Mutual exclusion and locks
Non-blocking data structures

Register Implementations
Safe, regular and atomic registers
General and limited transactions
Atomic snapshots

Hierarchy of objects
The FLP impossibility
The consensus number
Universal constructions
Transactional memories
Transactional algorithms
Opacity and obstruction-freedom

Keywords

Concurrency, parallelism, algorithms, data structures

Learning Prerequisites

Required courses

ICC, Operatings 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

Midterm and final exam
Project

Assessment methods

With continuous control, midterm final exams and project

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

Concurrent Algorithms, R. Guerraoui and P. Kouznetsov

Websites

- http://lpd.epfl.ch/site/education

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.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
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

• Algebra (MATH-310)
• Probabilities and statistics (MATH-232)
• Algorithms (CS-250)

Recommended courses

• Computer security (COM-301)

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

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.

Ressources en bibliothèque

- A computational introduction to number theory and algebra / Shoup
- Communication security / Vaudenay

Websites

- http://lasec.epfl.ch/teaching.shtml

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=13671

Prerequisite for

- Advanced cryptography (COM-401)
- Student seminar: security protocols and applications (COM-506)

EPFL

COM-480 **Data visualization**

Vuillon Laurent Gilles Marie

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.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Learning Sciences		Obl.
SC master EPFL	MA2, MA4	Opt.

Language	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

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-305 Software engineering (BA)
CS-250 Algorithms (BA)
CS-401 Applied data analysis (MA)

Recommended courses

EE-558 A Network Tour of Data Science (MA)
CS-486 Interaction design (MA)
CS-210 Functional programming (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%)

Supervision

Office hours	No
Assistants	No
Forum	No

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

- Interactive Data Visualization for the Web / Murray
- Visualization Analysis and Design / Munzner
- Data Visualisation / Kirk
- The Truthful Art / Cairo

Notes/Handbook

Lecture notes

Websites

- https://www.kirellbenzi.com

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15487

EPFL

CS-422 **Database systems**

Ailamaki Anastasia

Cursus	Sem.	Type
Computer and Communication Sciences		Obl.
Computer science minor	E	Opt.
Computer science	MA2, MA4	Obl.
Cybersecurity	MA2, MA4	Obl.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	7
Session	Summer
Semester	Spring
Exam	Written
Workload	210h
Weeks	14
Hours	**7 weekly**
Lecture	3 weekly
Exercises	2 weekly
Project	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. It covers a wide range of topics and technologies, and will prepare students to be able to build such systems as well as read and understand recent research publications.

Content

- Database systems
- Online analytics; data stream processing
- Column stores
- Decision support systems and data warehouses
- Large-scale data analytics infrastructure and systems
- Transaction processing. OLTP systems and concurrency control algorithms
- Distributed data management systems
- Query optimization; database tuning
- Logging and recovery
- Modern storage hierarchies

Learning Prerequisites

Required courses

- CS-107: Introduction to programming
- CS-322: Introduction to database systems
- CS-323: Introduction to operating systems
- CS-452: Foundations of software

Recommended courses

Learning Outcomes

By the end of the course, the student must be able 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 model and on the expected query workload
- Compare concurrency control algorithms, and algorithms for distributed data management
- Identify performance culprits, e.g., due to concurrency control

Teaching methods

Lectures, project, homework, exercises

Expected student activities

During the semester, the students are expected to:

- attend the lectures in order to ask questions and interact with the professor,
- attend the exercise sessions to solve and discuss exercises about the recently taught material,
- work on projects, which cover the practical side of the taught material,
- take a midterm,
- take a final exam,
- read scientific papers related to the course material

Assessment methods

- project
- exams

Supervision

Office hours	Yes
Others	Office hours on request. Q&A sessions in lectures and exercises.

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

- Database Management Systems / Ramakrishnan
- Mining of Massive Datasets / Rajaraman
- Readings in Database Systems / Hellerstein

EPFL

CS-438 **Decentralized systems engineering**

Ford Bryan Alexander

Cursus	Sem.	Type
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
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 development and testing of their own decentralized system incorporating messaging, encryption, and blockchain concepts.

Content

Topics this course covers include:

- Addressing, Forwarding, Routing. Peer-to-peer communication.
- Information gossip. UseNet: technical, security, and social lessons. Randomized rumor-mongering and anti-entropy algorithms.
- Communicating Securely: Basic Cryptographic Tools. Symmetric-key encryption. Hash functions, message authentication. Diffie-Hellman key exchange. Public-key encryption, digital signatures.
- Trust and Reputation. Authorities, trust networks. Sybil attacks and defenses.
- Naming and search. Request flooding. Hierarchical directories, landmark structures. 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.

Keywords

distributed systems, decentralized systems, security, privacy, anonymity, cryptography, gossip, consensus, swarming, blockchain, cryptocurrency

Learning Prerequisites

Required courses

- COM-208Computer networks

Recommended courses

- CS-206 Parallelism and concurrency
- COM-301 Computer security
- CS-323 Introduction to operating systems

Important concepts to start the course

Students must already be highly competent at programming and debugging in a high-level systems programming language such as Java, C#, or Go. Programming exercises will be in Go, but students already well-versed and experienced in programming with comparable systems languages should be able to pick up Go during the course.
Students should have both solid foundational knowledge of how networks function, and some experience actually writing network programs, e.g., TCP/IP programming using the Sockets API.

Learning Outcomes

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

- Implement systems via hands-on coding, debugging, and interoperability testing
- Design practical distributed and decentralized systems

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, social trust networks, 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 optionally in small teams during the final project-oriented part of the course.

Assessment methods

- Lecture attendance: 10%
- Programming assignment grading (evaluating both function and student documentation): 50%
- Final project grading (accounting for both scope, appropriateness, and follow-through in implementation quality and documentation): 40%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15483

EE-559

Deep learning

Fleuret François

Cursus	Sem.	Type
Civil & Environmental Engineering		Opt.
Computational science and Engineering	MA2, MA4	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		Obl.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	**500**

Summary

The objective of this course is to provide a complete introduction to deep machine learning. How to design a neural network, how to train it, and what are the modern techniques that specifically handle very large networks.

Content

The course aims at providing an overview of existing processings and methods, at teaching how to design and train a deep neural network for a given task, and at providing the theoretical basis to go beyond the topics directly seen in the course.

It will touch on the following topics:

- What is deep learning, introduction to tensors.
- Basic machine-learning, empirical risk minimization, simple embeddings.
- Linear separability, multi-layer perceptrons, back-propagation.
- Generalized networks, autograd, batch processing, convolutional networks.
- Initialization, optimization, and regularization. Drop-out, batchnorm, resnets.
- Deep models for Computer Vision.
- Analysis of deep models.
- Auto-encoders, embeddings, and generative models.
- Recurrent and attention models, Natural Language Processing.

Concepts will be illustrated with examples in the PyTorch framework (http://pytorch.org).

Keywords

machine learning, neural networks, deep learning, computer vision, python, pytorch

Learning Prerequisites

Required courses

• Linear algebra (vector, matrix operations, Euclidean spaces).
• Differential calculus (Jacobian, Hessian, chain rule).
• Python programming.
• Basics in probabilities and statistics (discrete and continuous distributions, normal density, law of large numbers, conditional probabilities, Bayes, PCA)

Recommended courses

• Basics in optimization (notion of minima, gradient descent).
• Basics in algorithmic (computational costs).
• Basics in signal processing (Fourier transform, wavelets).

Teaching methods

Ex-cathedra with exercise sessions and mini-projects. Possibly invited speakers.

Assessment methods

Mini-projects by groups of students, and one final written exam.

Resources

Notes/Handbook

Not mandatory: http://www.deeplearningbook.org/

Websites

• https://fleuret.org/ee559/

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.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Project	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

Virtual desktop infrastructure (VDI)

No

Bibliography

G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw'Hill.

Ressources en bibliothèque

- Synthesis and Optimization of Digital Circuits / De Micheli

Notes/Handbook

Copies of the slides used for lectures will be given in class and posted.

Websites

- http://lsi-www.epfl.ch/dtis/

CS-411

Digital education & learning analytics

Dillenbourg Pierre, Jermann Patrick

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Digital Humanities	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Oral
Workload	120h
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 on this topic: do student actually learn due to technologies?

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.*

Keywords

learning, pedagogy, teaching, online education, MOOCs

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
- Apply machine learning methods to educational traces

Transversal skills

- Set objectives and design an action plan to reach those objectives.

Teaching methods

The course will combine participatory lectures with a project around learning analytics

Expected student activities

The project will include several milestones to be delivered along the semester.

Assessment methods

- Project + exam
- 50 / 50

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14248

CS-451

Distributed algorithms

Guerraoui Rachid

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science minor	H	Opt.
Computer science	MA1, MA3	Obl.
Cybersecurity	MA1, MA3	Obl.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	2 weekly
Practical work	1 weekly
Number of positions	

Summary

Computing is often distributed over several machines, in a local IP-like network, a cloud or in a P2P network. Failures are common and computations need to proceed despite partial failures of machines or communication links. The foundations of reliable distributed computing will be studied.

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

Midterm and final exams
Project

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

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

Websites

- http://lpdwww.epfl.ch/education

CS-423

Distributed information systems

Aberer Karl

Cursus	Sem.	Type
Biocomputing minor	H	Opt.
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
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.
Energy Management and Sustainability	MA1, MA3	Opt.
Environmental Sciences and Engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

This course introduces the key concepts and algorithms from the areas 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. Association Rule Mining
2. Document Classification (knn, Naive Bayes, Fasttext, Transformer models)
3. Recommender Systems (collaborative filtering, matrix factorization)
4. 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
9. Data Integration

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 exercises (Python)

Assessment methods

25% Continuous evaluations with bonus system during the semester
75% Final written exam (180 min) during exam session

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

EPFL

ENG-466

Distributed intelligent systems

Martinoli Alcherio

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Civil Engineering	MA2, MA4	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Energy Management and Sustainability	MA2, MA4	Opt.
Energy Science and Technology	MA2, MA4	Opt.
Environmental Sciences and 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	English
Credits	5
Session	Summer
Semester	Spring
Exam	Oral
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	3 weekly
Number of positions	

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

Artificial intelligence, swarm intelligence, distributed robotics, sensor networks, modeling, machine-learning, control

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 homework in team

Expected student activities

Attending lectures, carrying out exercises and the course project, and reading handouts.

Assessment methods

Oral exam (60%) with continuous assessment during the semester (40%).

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Lecture notes, selected papers and book chapters distributed at each lecture.

Websites

- https://disal.epfl.ch/teaching/distributed_intelligent_systems/

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15472

Prerequisite for

R&D activities in engineering

COM-502

Dynamical system theory for engineers

Thiran Patrick

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.
Systems Engineering minor	E	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

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).
- 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.

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

• http://moodle.epfl.ch/course/view.php?id=303

Prerequisite for

Classes using methods from dynamical systems.

CS-473

Embedded systems

Beuchat René

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	English
Credits	4
Session	Winter
Semester	Fall
Exam	Oral
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Project	2 weekly
Number of positions	

Summary

The main topics of this course are understanding and designing embedded system on a programmable circuit (FPGA). Students will be able to design a camera or a LCD controller on an FPGA in VHDL and will use their controller through a softcore processor.

Content

- Microcontrollers and their associated programmable interfaces (GPIO, Timer, SPI, A/D, PWM, interrupts)
- Hardcore/softcore processors (ie. NIOS II, ARM)
- Memory organizations, little/big endian
- Synchronous busses, dynamic bus sizing (ie. Avalon Bus in Memory Mapped mode)
- Processor busses, busses realized in a FPGA
- Serial busses(ie. UART, SPI, i2c, ...)
- How an LCD graphical screen and a CMOS camera work
- FPGA-based conception of Embedded Systems
- Embedded systems with processors on FPGAs

Laboratories provide knowledge & practice to develop an embedded system based on an FPGA device.

Keywords

microprocessors, microcontroller, FPGA, embedded systems, SoC, programmable interface

Learning Prerequisites

Required courses

Introduction to computing systems, Logic systems, Computer architecture

Recommended courses

Electronic, Programming (C/C++), Project System On Chip

Important concepts to start the course

- Computer architecture (processor, memory, programmable interfaces)
- Processor Architecture (PC, registers, ALU, instruction decoding, instruction execution)
- Knowledge of C programming language

• Knowledge of VHDL

Learning Outcomes

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

- Design an embedded system on an FPGA
- Analyze a specific problem to be solved and propose an FPGA-based system to solve it
- Implement a solution to the given problem
- Realize and simulate the design
- Test the developed solution on an FPGA
- Use complex development tools and hardware debugging tools such as a logic analyzer and an oscilloscope

Transversal skills

- Use a work methodology appropriate to the task.
- Negotiate effectively within the group.
- Set objectives and design an action plan to reach those objectives.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Use both general and domain specific IT resources and tools
- Make an oral presentation.

Teaching methods

Ex cathedra and exercises, laboratories by specific sub-topics, final mini-project

Expected student activities

- Reading and deepening of course concepts
- Preparation of exercises performed in the laboratory
- Writing reports on different labs
- Realization of a final mini-project by group with oral presentation, report and demonstration

Assessment methods

With continuous control.
all labs 30%, mini-projet 20%, oral exam 50%

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Course on Moodle with forum

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Teaching notes and suggested reading material on moodle
Specialized datasheets (micro-controllers, FPGA) and standards(ie, SPI, i2c, Amba, Avalon, etc)

Notes/Handbook

Documents and slides provided on moodle

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=1231

Prerequisite for

CS-476 Real-time embedded systems

EPFL

COM-597

Engineering internship credited with Master project (master in Communication Systems)

Profs divers *

Cursus	Sem.	Type
SC master EPFL	MA1, MA2, MA3, MA4, PME, PMH	Opt.

Language	English
Credits	0
Session	Winter, Summer
Semester	Fall
Exam	Term paper
Workload	0h
Weeks	
Practical work	320 weekly
Number of positions	

Remark

L'étudiant doit effectuer un stage de 8 semaines pendant l'été ou de 6 mois après un semestre de Master.

Summary

The engineering internship is part of the curriculum for master's students. They join companies in Switzerland or abroad for an internship or a master's thesis project which takes place in a field of activity where the skills of the future engineer are highlighted.

Content

Internships represent an important experience for students, allowing them to achieve the following goals:
- Immerse themselves in the professional world
- Highlight the importance of teamwork
- Consider the imperatives of a company in its processes
- Put into practice the knowledge acquired from the study plan

The following three forms of internship are possible as part of the master's study plan :
- 8 weeks internship during the summer only
- 6 months (long) internship. The student is on leave for one semester
- Master's project in industry (25 weeks)

Learning Prerequisites

Required courses

- Have completed one full semester for students who have obtained their Bachelor at EPFL
- Have completed two full semesters for students coming from another university

Important concepts to start the course

Importance of commitment

Learning Outcomes

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

- Be aware of the importance of legal procedures and the ethical code of the profession
- Communicate effectively and be understood
- Self-assess the levef acquired skills and plan the next goals
- Manage the priorities

- 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/

EPFL

CS-491 Enterprise and service-oriented architecture

Regev Gil

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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 implicit wishes.

Content

Target Audience

EPFL 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 and business 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 becoming the most important enabler of enterprise strategy, these roles are becoming 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:

- The nature of organizations
- Problems and solutions
- Requirements elicitation
- Enterprise modeling
- Low-code prototyping
- Creating a request for tender

Keywords

Ethnography, interviews, contextual inquiry, business service, business process, IT service, business analysis, requirements engineeing, SEAM service modeling, SEAM motivation modeling, interpretivism, homeostasis, appreciation, resilience, low-code development, request for tender

Learning Outcomes

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

- Elicit requirements with business stakeholders
- Analyze business stakeholder perceptions and motivations

- Assess / Evaluate business processes
- Define requirements for business and IT services
- Present problems and solutions to management
- Implement a prototype in a low-code platform

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

Experiential learning and teamwork

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.

Ressources en bibliothèque

- Contextual design / Beyer

CS-489

Experience design

Huang Jeffrey

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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	

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

Workshops, 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

Bibliography

To be made available during the course

EPFL

CS-550 **Formal verification**

Kuncak Viktor

Cursus	Sem.	Type
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Practical work	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 and Predicate Abstraction
- Information Flow and Taint Analysis
- Verification of Security Protocols
- Dependent and Refinement Types

Learning Prerequisites

Recommended courses

Computer Language Processing / Compilers

Important concepts to start the course

Discrete Mathematics

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 material and complete and explain projects during the semester.

Assessment methods

The grade is based on the code, documentation, and explanation of projects during the semester.
There are no written exams.

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 model checking / Clarke

- Introduction to mathematical logic and type theory / Andrews
- Principles of Program Analysis / Flemming
- The Calculus of Computation / Bradley
- Logic in Computer Science / Huth
- Handbook of model checking : Model Checking Security Protocols / Bassin

Notes/Handbook

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

Websites

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

Videos

- https://tube.switch.ch/channels/f2d4e01d

Prerequisite for

MSc thesis in the LARA group

EPFL

CS-457

Geometric computing

Pauly Mark

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**5 weekly**
Lecture	3 weekly
Practical work	2 weekly
Number of positions	

Summary

This course will cover mathematical concepts and efficient numerical methods for geometric computing. We will develop and implement 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 theory and implementation homeworks

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

EPFL

MATH-483 Gödel and recursivity

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	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 (pas donné en 2021-22)

Summary

Gödel incompleteness theorems and mathematical foundations of computer science

Content

Gödel's theorems:
Peano and Robinson Arithmetics. Representable functions. Arithmetic of syntax. Incompleteness, and undecidability theorems.

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.

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 :

EPFL

- 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
- Introduction to Set theory / Hrbacek
- 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
- Inexhaustibility, a non exhaustive treatment / Franzen

Websites

- http://www.hec.unil.ch/logique/enseignement/recursivity

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=14569

EPFL

MICRO-511

Image processing I

Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Biocomputing minor	H	Opt.
Computational Neurosciences minor	H	Opt.
Computational science and Engineering	MA1, MA3	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.
Neuroprosthetics minor	H	Opt.
Photonics minor	H	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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 in JAVA. 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. Linear and Max-Lloyd Quantization.
- Characterization of discrete images and linear filtering. z-transform. Convolution. Separability. FIR and IIR filters.
- Image-processing operations. Point operators (thresholding, histogram modification). Spatial operators (smoothing, enhancement, nonlinear filtering). Morphological operators.
- Introduction to image analysis and computer vision. Segmentation, edge detection, objet detection, image comparison.

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

Transversal skills

- Use a work methodology appropriate to the task.
- Manage priorities.
- Use both general and domain specific IT resources and tools

EPFL

MICRO-512

Image processing II

Liebling Michael, Sage Daniel, Unser Michaël, Van De Ville Dimitri

Cursus	Sem.	Type
Biocomputing minor	E	Opt.
Computational Neurosciences minor	E	Opt.
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Life Sciences Engineering	MA2, MA4	Opt.
Microtechnics	MA2, MA4	Opt.
Neuroprosthetics minor	E	Opt.
Photonics minor	E	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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 JAVA; application to real-world examples in industrial vision and biomedical imaging.

Content

- **Review of fundamental notions.** Multi-dimensional Fourier transform. Convolution. z-transform. Digital 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 from projections.** X-ray scanners. Radon transform. Central slice theorem. Filtered backprojection. Iterative methods.
- **Deconvolution.** Inverse and Wiener filtering. Matrix formulations. Iterative techniques (ART).
- **Statistical pattern classification.** Decision making. Bayesian classification. Parameter estimation. Supervised learning. Clustering.
- **Image analysis.** Pixel classification. Contour extraction and representation. Shape. Texture. Snakes and active contours.

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.)

Learning Outcomes

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

- 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

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

EPFL

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.
Microtechnics	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Oral
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Project	1 weekly
Number of positions	

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) 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, ...)
6. Real-time response and performance analysis

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.

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 based on availability)
- Work on group assignment
- Hand-in artifacts for assignment on time

Assessment methods

Assignment 25% and final oral exam 75%

Resources

Bibliography

Nussbaumer, Informatique Industrielle (EPFL)
Olsson, Gustav & Rosen, Christian - industrial automation,Dept. of Industrial Electrical Engineering and Automation, Lund University, Lund, Sweden.
Introduction to Industrial Automation, Stamatios Manesis & George Nikolakopoulos, CRC Press, 2018

Ressources en bibliothèque

- Informatique Industrielle / Nussbaumer

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=14114

COM-402

Information security and privacy

Hubaux Jean-Pierre, Pyrgelis Apostolos

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	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.
Data science minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
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 Big Data world. The tools are illustrated with relevant applications.

Content

- Overview of cyberthreats
- Exploiting 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 Python programming or better
Basic networking knowledge

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
- Understand the key technical tools available for security/privacy protection

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

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.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
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

Ressources en bibliothèque

- Elements of Information Theory / Cover

Websites

- http://moodle.epfl.ch/enrol/index.php?id=14593

EPFL

CS-430 **Intelligent agents**

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.
Energy Management and Sustainability	MA1, MA3	Opt.
Financial engineering minor	H	Opt.
Financial engineering	MA1, MA3	Opt.
Learning Sciences		Obl.
Robotics, Control and Intelligent Systems		Opt.
Robotics	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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	

Remark

pas donné en 2021-22

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

Mini-projects and exercises 40%, final exam 60%

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 Edition), Prentice Hall Series in Artificial Intelligence, 2003/2009.

Ressources en bibliothèque

- Artificial Intelligence: A Modern Approach / Russell
- An Introduction to MultiAgent Systems / Wooldridge

Websites

- http://liawww.epfl.ch/
- http://moodle.epfl.ch/

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	English
Credits	4
Session	Summer
Semester	Spring
Exam	During the semester
Workload	120h
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 to be carried out by a team of three students.

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, design for playfulness, rapid prototyping techniques, evaluation with users.

Learning Prerequisites

Required courses

Introduction to Visual Computing

Recommended courses

Open to students enrolled in the Master and PhD programs in IC.

Important concepts to start the course

Goal-direction design

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 intearctions
- 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, exercises, hands-on practice, design review

Expected student activities

Lectures, readings, design project, quiz

Assessment methods

Group project, presentation, mid-term exam

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

CS-431

Introduction to natural language processing

Chappelier Jean-Cédric, Rajman Martin

Cursus	Sem.	Type
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		Obl.
SC master EPFL	MA1, MA3	Opt.
UNIL - Sciences forensiques	H	Opt.

Language	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 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: (1) morpho-lexical level: electronic lexica, spelling checkers, ...; (2) syntactic level: regular, context-free, stochastic grammars, parsing algorithms, ...; (3) semantic level: models and formalisms for the representation of meaning, ...
Several application domains will be presented: Linguistic engineering, Information Retrieval, Text mining (automated knowledge extraction), Textual Data Analysis (automated document classification, visualization of textual data).

Keywords

Natural Language Processing; Computationnal Linguisitics; Part-of-Speech tagging; Parsing

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, parsing)
- Describe the typical problems and processing layers in NLP
- Analyze NLP problems to decompose them in adequate independant components

Teaching methods

Ex cathedra ; practical work on computer

Expected student activities

attend lectures and practical sessions, answer quizzes.

Assessment methods

4 quiz during semester 25%, final exam 75%

Supervision

Office hours	No
Assistants	No
Forum	No

Resources

Virtual desktop infrastructure (VDI)

No

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
- Speech and Language Processing / Jurafsky
- Speech and Language Engineering / Rajman
- Foundations of Statistical Natural Language Processing / Manning

Websites

- http://coling.epfl.ch

CS-526

Learning theory

Macris Nicolas, Urbanke Rüdiger

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

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) dimension, non-uniform learnability, complexity of learing.
- Neural Nets : representation power of neural nets, learning and stability, PAC Bayes bounds.
- Graphical model learning.
- Non-negative matrix factorization, Tensor decompositions and factorization.
- Learning mixture models.

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 and non-uniform learnability
- Describe basic facts about representation of functions by neural networks
- Describe recent results on specific topics e.g., graphical model learning, 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
Biocomputing minor	H	Opt.
Civil & Environmental Engineering		Opt.
Communication systems minor	H	Opt.
Computational Neurosciences minor	H	Opt.
Computational science and Engineering	MA1, MA3	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		Obl.
Life Sciences Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Obl.

Language	English
Credits	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
Hours	**6 weekly**
Lecture	4 weekly
Exercises	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 (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 models: Regression, classification, clustering, dimensionality reduction, neural networks, time-series analysis
- 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

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

- Linear algebra and learning from data
- The elements of statistical learning : data mining, inference, and prediction / Friedman
- Understanding Machine Learning / Shalev-Shwartz
- Neural Networks and Deep Learning / Nielsen
- Machine Learning: A Probabilistic Perspective / Murphy
- Pattern Recognition and Machine Learning / Bishop

Notes/Handbook

https://github.com/epfml/ML_course

Websites

- https://www.epfl.ch/labs/mlo/machine-learning-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.
Digital Humanities	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

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-233a / CS-233b 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 disadvantages and explain the differences 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%)

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

EPFL

COM-516

Markov chains and algorithmic applications

Lévêque Olivier, Macris Nicolas

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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 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

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.

COM-514

Mathematical foundations of signal processing

Bejar Haro Benjamin, Fageot Julien René Pierre, Simeoni Matthieu

Cursus	Sem.	Type
Communication systems minor	H	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer and Communication Sciences		Opt.
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA1, MA3	Opt.
Systems Engineering minor	H	Opt.

Language	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

Signal processing tools are presented from an intuitive geometric point of view which is at the heart of all modern signal processing techniques. The student will develop the mathematical depth and rigor needed for the study of advanced topics in signal processing and approximation theory.

Content

Sequences, Discrete-Time Systems, Functions and Continuous-Time Systems (review of discrete-time Fourier transform; DFT; Fourier transform and Fourier series).
From Euclid to Hilbert: Linear Algebra Fundamentals for Representation Theory (vector spaces; Hilbert spaces; approximations, projections and decompositions; bases and frames; linear operators; adjoint; generalized inverses; matrix representations; computational aspects)
Sampling and Interpolation (sampling and interpolation with normal and non orthogonal vectors, sequences and functions; sampling and interpolation of bandlimited sequences and functions)
Polynomial and Spline Approximation (Legendre and Chebyshev polynomials; Lagrange interpolation; minimax approximation; Taylor expansions; B-splines)
Regularized Inverse Problems (regularized convex optimisation; Tikhonov regularisation; penalised basis pursuit; proximal algorithms; pseudo-differential operators and L-splines; representer theorems for continuous inverse problems with Tikhonov penalties)

Learning Prerequisites

Required courses

Signal processing for communications (or Digital signal processing on Coursera)
Linear Algebra I and II (or equivalent).

Recommended courses

Signals and Systems

Important concepts to start the course

Good knowledge of linear algebra concepts. Basics of Fourier analysis and signal processing. Good knowledge of Python and its scientific packages (Numpy, Scipy).

Learning Outcomes

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

- Master the right tools to tackle advanced signal and data processing problems
- Develop an intuitive understanding of signal processing through a geometrical approach
- Get to know the applications that are of interest today
- Learn about topics that are at the forefront of signal processing research
- Identify and implement the algorithm best suited to solve a given convex optimisation problem
- Assess the computational cost and numerical stability of a numerical solver

Transversal skills

- Collect data.
- Write a scientific or technical report.
- Use a work methodology appropriate to the task.
- Demonstrate the capacity for critical thinking
- Use both general and domain specific IT resources and tools

Teaching methods

Ex cathedra with exercises, homeworks and practicals.

Expected student activities

Attending lectures, completing exercises.

Assessment methods

homeworks and project assignement 50%, final exam (written) 50%

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

M. Vetterli, J. Kovacevic and V. Goyal, *"Signal Processing: Foundations"*, Cambridge U. Press, 2014.
Available in open access at http://www.fourierandwavelets.org

Ressources en bibliothèque

- Signal Processing: Foundations / Vetterli

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=13431

COM-405

Mobile networks

Hubaux Jean-Pierre

Cursus	Sem.	Type
Communication systems minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Obl.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

This course provides a detailed description of the organization and operating principles of mobile communication networks.

Content

Introduction to wireless networks
Organization of the MAC layer
Wireless Local Area Networks - WiFi
Cellular networks
Mobility at the network and transport layers
Security and privacy in mobile networks

Keywords

Communication networks, protocols, mobility

Learning Prerequisites

Required courses

COM-208 Computer Networks

Recommended courses

COM-302 Principles of Digital Communications
COM-301 Computer security

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, bottlenecks and vulnerabilities

Teaching methods

Ex cathedra lectures
Weekly quizzes
Exercise sessions
Hands-on exercises

Expected student activities

Class participation, quizzes, homework, hands-on exercises

Assessment methods

Quizzes + final exam

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

Handouts, recommended books (see course URL)

Ressources en bibliothèque

- Fundamentals of Mobile Data Networks / Miao

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	English
Credits	6
Session	Winter
Semester	Fall
Exam	During the semester
Workload	180h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Practical work	2 weekly
Number of positions	

Summary

This course complements the theoretical knowledge learned in PDC with more advanced topics such as OFDM, MIMO, fading chancels, 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 in Principles of Digital Communications.
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 communication 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 and advanced digital communication system (data rate, spectral bandwidth, energy requirements, error probability, implementation complexity)
- Model 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

COM-512

Networks out of control

Grossglauser Matthias, Thiran Patrick

Cursus	Sem.	Type
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Génie électrique		Opt.
Informatique	MA2, MA4	Opt.
Mineur en Data science	E	Opt.
Mineur en Systems Engineering	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	4
Session	Summer
Semester	Spring
Exam	Written
Workload	120h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal

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%.

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

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
- Random Graph Dynamics / Durrett
- Networks, Crowds and Markets / Easley
- Poisson Approximation / Barbour
- Percolation / Grimmett

Notes/Handbook

Class notes will be available on the course website.

Websites

- http://icawww1.epfl.ch/class-nooc/

EPFL

MATH-489 **Number theory in cryptography**

Jetchev Dimitar

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

Much of the course draws inspiration from the Math-489 (-2019) curriculum taught by Prof. Dimitar Jetchev.

Keywords

public key cryptography, key exchange, digital signatures, zero knowledge proofs, RSA, ElGamal, integer factorization, index calculus, elliptic curve cryptography

Teaching methods

lectures, exercises, additional references

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 no graded midterm since the class is online. 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.

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.

EPFL

CS-439

Optimization for machine learning

Flammarion Nicolas, Jaggi Martin

Cursus	Sem.	Type
Computational science and Engineering	MA2, MA4	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Obl.
Data science minor	E	Opt.
Electrical Engineering		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	2 weekly
Practical work	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

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Websites

- https://github.com/epfml/OptML_course

Videos

- https://www.youtube.com/playlist?list=PL4O4bXkI-fAeYrsBqTUYn2xMjJAqlFQzX

EPFL

COM-507

Optional project in communication systems

Profs divers *

Cursus	Sem.	Type
Communication systems minor	E, H	Opt.
SC master EPFL	MA1, MA2, MA3, MA4	Opt.

Language	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 authorization of the section. Only for 2nd year Master students. Supervision by an IC professor

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 :

https://www.epfl.ch/schools/ic/communication-systems-msc/projets-laboratory-msc/

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.

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

Virtual desktop infrastructure (VDI)

No

Websites

- https://www.epfl.ch/schools/ic/education/master/semester-project-msc/

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	English
Credits	7
Session	Winter
Semester	Fall
Exam	During the semester
Workload	210h
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:

- COM-208 Computer networks
- CS-208/209 Computer architecture
- CS-210 Functional programming
- CS-305 Software engineering
- CS-322 Introduction to database systems
- CS-323 Introduction to operating 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

- 30% written assignments (during the semester)
- 10% research presentation (during the semester)
- 10% course participation and discussion (during the semester)
- 50% written final exam (during the exam session)

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

MATH-467

Probabilistic methods in combinatorics

Marcus Adam W.

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	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

We develop a sophisticated framework for solving problems in discrete mathematics through the use of randomness (i.e., coin flipping). This includes constructing mathematical structures with unexpected (and sometimes paradoxical) properties for which no other methods of construction are known.

Content

- Linearity of expectation
- The second moment method
- Local lemma
- Random graphs and matrices
- Applications in combinatorics and graph theory

Keywords

random variable, expected value, probabilistic method, random graph

Learning Prerequisites

Required courses

Probability theory

Recommended courses

- Discrete Mathematics or Graph Theory
- Linear Algebra

Important concepts to start the course

Graph, random variable, expectation, variance, binomial coefficients, asymptotics, eigenvalues

Learning Outcomes

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

- Define and explain basic concepts in probability and discrete mathematics
- Prove explain, and apply the first and second moment methods

- Prove explain, and apply the Local Lemma
- Solve exercises, design randomized algorithms
- Describe and explain the method of interlacing polynomials

Transversal skills

- Summarize an article or a technical report.
- Demonstrate the capacity for critical thinking
- Assess progress against the plan, and adapt the plan as appropriate.

Teaching methods

Lectures and exercises

Expected student activities

Attending the lectures, solving the exercises, reading sections from the textbook

Assessment methods

Exam written
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.

Resources

Bibliography

Noga Alon-Joel Spencer: The Probabilistic Method (Wiley)

Ressources en bibliothèque

- Noga Probabilistic method

EPFL

CS-476 **Real-time embedded systems**

Beuchat René

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	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

A real time system is subject to important temporal constraints. This course is about understanding where processing time is spent and what a designer can do in order to achieve real-time processing systems. Some solutions are Multiprocessors, accelerators, custom instructions, specialized hardware.

Content

During this course, response time measurements of interrupts are studied in laboratories, such as for example: the influence of dynamic memories, cache memories, compilation flags. Interrupts response time measurements, task commutations and synchronizations primitives are carried out on an embedded system based on an FPGA.
The course includes the study of embedded systems management models through polling, interrupts and using a real time kernel with its task management and synchronization primitives.
Specialized programmable interfaces are implemented in VHDL to help with these measurements. A real time kernel is studied and used during the labs. An acquisition system is implemented and the gathered data is transmitted by a Web server. To ensure the real time acquisition and reading by the Web server, a multiprocessor system is developed and implemented on an FPGA.
An Accelerator designed in VHDL makes it possible to facilitate the optimization of functions through hardware on an FPGA. Cross development tools are used.
Each topic is treated by a theoretical course and an associated laboratory. The laboratories are realized on an FPGA board including a hardcore multiprocessor. A real time operating system is studied and used with the laboratories.

Keywords

Real Time, FPGA, SOC, microprocessor, hardware accelerator, custom instruction, Real Time OS

Learning Prerequisites

Required courses

Introduction to computing systems, Logic systems, Computer architecture

Recommended courses

Embedded Systems, Real time Programming

Important concepts to start the course

Programmable Logic Architecture (FPGA), Computer Architecture, VHDL, C programming, Real Times basic knowledge (semaphor, synchronization)

Learning Outcomes

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

- Design a multiprocessor system on an FPGA
- Analyze the performance of a real time embedded system
- Use design tools for SOC conception on an FPGA
- Implement a complete real-time system based on a multiprocessor design on an FPGA
- Test the realized system
- Defend the choices during the design phases

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Communicate effectively, being understood, including across different languages and cultures.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Make an oral presentation.
- Write a scientific or technical report.

Teaching methods

Ex cathedra, laboratories and a mini-project

Expected student activities

- 3 groups of laboratories on specific topics, with a report by group for each of them, 3-4 weeks/topic;
- A final mini-project to practically synthesize the content of the course, with the design of a multiprocessor system on an FPGA, including for example a Web-server, a camera controller, a specific algorithm to be implemented in an FPGA hardware accelerator, 3~4 weeks for this mini-project

Assessment methods

Continuous control with reports and oral presentation
all labs 50% + final mini-project 50%

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

Teaching notes and suggested reading material.
Specialized datasheets (ie.ex. FPGA et specific microcontrollers) and standards.

Notes/Handbook

Slides and documents on moodle

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=391

COM-416

Semester project in communication systems II

Profs divers *

Cursus	Sem.	Type
SC master EPFL	MA1, MA2, MA3, MA4	Obl.

Language	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 :
https://www.epfl.ch/schools/ic/communication-systems-msc/projets-laboratory-msc/

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

Virtual desktop infrastructure (VDI)

No

Websites

- https://www.epfl.ch/schools/ic/education/master/semester-project-msc/

EE-511

Sensors in medical instrumentation

Aminian Kamiar

Cursus	Sem.	Type
Biomedical technologies minor	E	Opt.
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Life Sciences 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	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	2 weekly
Exercises	1 weekly
Number of positions	

Summary

Fundamental principles and methods used for physiological signal conditioning. Resistive, capacitive, inductive, piezoelectric and optical techniques used to detect and convert physiological information's to electrical signals. Laboratory and ambulatory devices for monitoring and therapy.

Content

1. **Physiological Mesurands**
Biopotentials; bioimpedance; mechanical, acoustic and thermal signals
2. **Noise in medical instrumentation**
Source and nature of the noise; noise reduction; instrumentation amplifier for biopotential measurement
3. **Biopotential measurement**
Electrodes; ECG, EMG and EEG measurement
4. **Resistive sensors**
Thermistor and its biomedical applications; strain gage for the measurement of blood pressure; force and accelerations of the body
5. **Inductive sensors**
Simple and mutual inductance and its medical applications
6. **Capacitive sensors**
Respiratory flow measurement by the gradient of pressure
7. **Piezoelectric sensors**
Force platform, accelerometer, angular rate sensor for the measurement of tremors and body movements, ultrasound transducer : measurement of pressure and flow rate
8. **Optical sensors**
Photoplethysmography; pulsed oxymetry
9. **Example of applications**

Keywords

sensors, instrumentation, biomedical devices, physiological measurement, monitoring

Learning Prerequisites

Required courses

courses en electrical circuit, basic electronics

Recommended courses

measuring systems or electronics or sensors

Important concepts to start the course

basic electronics, basic physics

Learning Outcomes

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

- Choose techniques detecting and convert physiological information's to electrical signals
- Exploit fundamental principles and methods used for physiological signal conditioning
- Design measuring devices
- Interpret error, noise in biomedical measuring systems

Transversal skills

- Use a work methodology appropriate to the task.
- Communicate effectively with professionals from other disciplines.

Teaching methods

Ex cathedra, with exercises

Expected student activities

home work, short quizzes during semester

Assessment methods

Written

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Bibliography

Medical Instrumentation : Application and design, JG Webster

Ressources en bibliothèque

- Medical Instrumentation / Webster

Notes/Handbook

Slides copies (to be completed during the lectures)
Polycopies (in French only)

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=2571

Prerequisite for

Semester project and Master project

EPFL

MATH-318

Set theory

Duparc Jacques

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

Set Theory as a foundational system for mathematics. ZF, ZFC and ZF with atoms. Relative consistency of the Axiom of Choice, the Continuum Hypothesis, the reals as a countable union of countable sets, the existence of a countable family of pairs without any choice function.

Content

Set Theory: ZFC. Extensionality and comprehension. Relations, functions, and well-ordering. Ordinals. Class and transfinite recursion. Cardinals. Well-founded relations, axiom of foundation, induction, and von Neumann's hierarchy. Relativization, absoluteness, reflection theorems. Gödel's constructible universe L. Axiom of Choice (AC), and Continuum Hypothesis inside L. Po-sets, filters and generic extensions. Forcing. ZFC in generic extensions. Cohen Forcing. Independence of the Continuum Hypothesis. HOD and AC: independence of AC. The reals without AC. Symmetric submodels of generic extensions. Applications of the symmetric submodel techinique (the reals as a countable union of countable sets, the reals not well-orderable, every ultrafilter on the integers is trivial). ZF with atoms and permutation models. Simultating permutation models by symmetric submodels of generic extensions.

Keywords

Set Theory, Relative consistency, ZFC, Ordinals, Cardinals, Transfinite recursion, Relativization, Absoluteness, Constructible universe, L, Axiom of Choice, Continuum hypothesis, Forcing, Generic extensions

Learning Prerequisites

Required courses

MATH-381 Mathematical Logic.
In particular ordinal numbers and ordinal arithmetic will be considered known and admitted.

Recommended courses

Mathematical logic (or any equivalent course on first order logic). Warning: without a good understanding of first order logic, students tend to get lost sooner orl later.

Important concepts to start the course

- 1st order logic
- basics of proof theory
- Basics of model theory
- Compacity theorem
- Löwenheim-Skolem

- Completeness theorem

Learning Outcomes

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

- Specify a model of ZFC
- Prove consistency results
- Develop a generic extension
- Argue by transfinite induction
- Decide whether ZFC proves its own consistency
- Formalize the axioms of ZF, AC, CH, DC
- Sketch an inner model
- Justify the axiom of foundation

Teaching methods

Ex cathedra lecture and exercises

Expected student activities

- Attendance at lectures
- Solve the exercises

Assessment methods

- Writen exam (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

1. Kenneth Kunen: Set theory, Springer, 1983
2. Lorenz Halbeisen: Combinatorial Set Theory, Springer 2018
3. Thomas Jech: Set theory, Springer 2006
4. Jean-Louis Krivine: Theorie des ensembles, 2007
5. Patrick Dehornoy: Logique et théorie des ensembles; Notes de cours, FIMFA ENS: http://www.math.unicaen.fr/~dehornoy/surveys.html
6. Yiannis Moschovakis: Notes on set theory, Springer 2006
7. Karel Hrbacek and Thomas Jech: Introduction to Set theory, (3d edition), 1999

Ressources en bibliothèque

- Introduction to Set theory / Hrbacek
- Set theory / Jech
- Logique et théorie des ensembles / Dehorny
- Set theory / Kunen
- Notes on set theory / Moschovakis
- Theorie des ensembles / Krivine

Notes/Handbook

Lecture notes (350 pages).

Websites

- http://www.hec.unil.ch/logique/

Moodle Link

- http://moodle.epfl.ch/course/index.php?categoryid=72

EPFL

EE-472

Smart grids technologies

Le Boudec Jean-Yves, Paolone Mario

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Obl.
Energy Management and Sustainability	MA2, MA4	Opt.
Energy Science and Technology	MA2, MA4	Opt.
Energy minor	E	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	5
Session	Summer
Semester	Spring
Exam	Written
Workload	150h
Weeks	14
Hours	**5 weekly**
Lecture	2 weekly
Exercises	1 weekly
Practical work	2 weekly
Number of positions	

Summary

Learn the technologies and methodologies used in the context smart electrical grids and be able to deploy/implement/test them in a lab environment.

Content

1. Modern monitoring: phasor measurement units technology, synchrophasors extraction processes and time alignement
2. Smart grid communication; reliability, real time and security issues
3. Topology assessment and contingency analysis of power grids
4. Admittance matrix calculus, numerical solution of the load flow problem and state estimation
5. Energy management and dispatch plans, the optimal power flow problem
6. Demand response

Keywords

Smart grid, power systems

Learning Prerequisites

Required courses

Electric power systems, power distribution networks, TPC/IP Networking

Recommended courses

Signal processing, discrete optimization methods, model predictive control, industrial electronics.

Important concepts to start the course

Understanding of electrical grids and communication networks.

Learning Outcomes

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

- Design monitoring and control platforms for smart grids
- Test a smart grid
- Implement a smart grid
- Analyze performance of a smart grid

Transversal skills

- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking
- Manage priorities.
- Use both general and domain specific IT resources and tools

Teaching methods

Ex cathedra, classroom integrated exercises and computer laboratory sessions.

Expected student activities

Attend lectures and labs
Do lab homeworks
Do online quizzes

Assessment methods

Written exam (50%) and graded lab reports (50%)

Prerequisite for

Master projects in the areas of power systems and energy conversion systems.

EE-593

Social media

Vonèche Cardia Isabelle, Gillet Denis

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Electrical and Electronical Engineering	MA2, MA4	Opt.
Learning Sciences		Obl.
SC master EPFL	MA2, MA4	Opt.
UNIL - HEC	E	Opt.

Language	English
Credits	2
Withdrawal	Unauthorized
Session	Summer
Semester	Spring
Exam	During the semester
Workload	60h
Weeks	14
Hours	**2 weekly**
Lecture	1 weekly
Project	1 weekly
Number of positions	**45**

Summary

The objective is to enable students to critically apprehend the Human Computer Interaction (HCI) challenges associated with the design and the exploitation of social media platforms.

Content

- Social media platforms and the long tail (definition and typology)
- Usability and adoption of social media platforms
- Web 2.0 features and adoption factors
- Privacy, trust and reputation models
- Identities, traces, and Web analytics
- Interplay, between platforms and communities (interdisciplinary perspective)
- Opportunities, requirements and constraints for organization and enterprises
- Participatory design methodologies
- Future ad hoc social applications

Learning Outcomes

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

- Choose
- Design
- Critique
- Defend

Transversal skills

- Set objectives and design an action plan to reach those objectives.
- Plan and carry out activities in a way which makes optimal use of available time and other resources.
- Use a work methodology appropriate to the task.
- Communicate effectively, being understood, including across different languages and cultures.

- Communicate effectively with professionals from other disciplines.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Negotiate effectively within the group.
- Assess one's own level of skill acquisition, and plan their on-going learning goals.

Teaching methods

Lectures, invited speakers, individual work and teamwork

Assessment methods

One individual project and one teamwork with combined peer and expert assesment (reports and presentations)

Supervision

Office hours	No
Assistants	Yes
Forum	No

Resources

Bibliography

- Chris Anderson (2006) - The Long Tail: Why the Future of Business is Selling Less of More. New York, NY: Hyperion. ISBN 1-4013-0237-8.
- Joshua Porter - Designing for the Social Web
- Matthew A. Russel - Mining the Social Web: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites. O¿Reilly 2011

Ressources en bibliothèque

- Designing for the Social Web / Porter
- Mining the Social Web / Russel
- The Long Tail / Anderson

EPFL

CS-412 **Software security**

Payer Mathias

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	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	3 weekly
Exercises	2 weekly
Practical work	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

Important concepts to start the course

Basic computer literacy like system administration, build systems, basic 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 may be peppered with occasional short surprise quizzes that are not mandatory but may earn points for successful participants. 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. They consist mostly of paper questions involving the analysis, critical review, and occasional correction of software. They include a reading, writing, and presentation assignment.

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 and assignments throughout the semester. The labs will account for 70%, the quizzes and assignments to 30%.

Supervision

Office hours	Yes
Assistants	Yes
Forum	No

Resources

Notes/Handbook

Software Security: Principles, Policies, and Protection (SS3P, by Mathias Payer)
http://nebelwelt.net/SS3P/

MATH-486

Statistical mechanics and Gibbs measures

Friedli Sacha

Cursus	Sem.	Type
Data Science	MA2, MA4	Opt.
Ing.-math	MA2, MA4	Opt.
Mathématicien	MA2	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

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.

Resources

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

- Gibbs Measures and Phase Transitions / Georgii
- (electronic version)
- Statistical mechanics of lattice systems / Friedli & Velenik

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	English
Credits	6
Session	Summer
Semester	Spring
Exam	Written
Workload	180h
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 multiple 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).

Recommended courses

Mathematical Foundations of Signal Processing (COM-514).

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.

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
- 10% mini project
- 70% 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 Probability Models / Ross
- 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 Hilbert Spaces with Applications / Debnath

Notes/Handbook

- Slides handouts;
- Collection of exercises.

Websites

- http://lcav.epfl.ch/cms/site/lcav/lang/en/teaching/statistical_sp_and_applications

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=422

MATH-413

Statistics for data science

Olhede Sofia Charlotta

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Data Science	MA1, MA3	Obl.
Data science minor	H	Opt.
Electrical Engineering		Opt.
Electrical and Electronical Engineering	MA1, MA3	Opt.
Managmt, tech et entr.	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	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

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
- All of Statistics.
- Statistics for Mathematicians
- Statistical Models

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=15506

EPFL

COM-506

Student seminar: security protocols and applications

Vaudenay Serge

Cursus	Sem.	Type
Cyber security minor	E	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	3
Session	Summer
Semester	Spring
Exam	Written
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
- do the final exam

Assessment methods

- lecture and attendance to others' lectures (50%)
- final exam (50%)

Supervision

Office hours	No
Assistants	Yes
Forum	Yes
Others	Lecturers and assistants are available upon appointment.

Resources

Websites

- http://lasec.epfl.ch/teaching.shtml

Moodle Link

- https://moodle.epfl.ch/course/view.php?id=13965

CS-448 Sublinear algorithms for big data analysis

Kapralov Mikhail

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
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

COM-407

TCP/IP networking

Le Boudec Jean-Yves

Cursus	Sem.	Type
Communication systems 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	English
Credits	6
Session	Winter
Semester	Fall
Exam	Written
Workload	180h
Weeks	14
Hours	**6 weekly**
Lecture	2 weekly
Exercises	2 weekly
Practical work	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

LECTURES: 1. The TCP/IP architecture 2. Layer 2 networking; Bridging. 3. The Internet protocol versions 4 and 6 4. The transport layer, TCP, UDP, sockets, QUIC. 5. Link state routing, OSPF, Distance Vector routing. Interdomain routing, BGP. 6. Congestion control principles. Application to the Internet. The fairness of TCP. Tunnels and hybrid architectures.
LABS: 1. Configuration of a network, virtual machines and mininet, packet captures 2. MAC; NATs and troubleshooting 3. Socket programming 4. OSPF routing 5. Congestion control and flow management 6. BGP

Keywords

TCP/IP
Computer Networks

Learning Prerequisites

Required courses

A first programming course

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 with questionnaires.
Online quizzes.
Labs on student's computer and if required and if possible, in the Internet Engineering Workshop

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 : Principles, Protocols and Practice", O. Bonaventure, open source textbook, http://inl.info.ucl.ac.be/CNP3

Ressources en bibliothèque

- Computer Networking / Bonaventure

Notes/Handbook

Slides are on moodle

Websites

- http://moodle.epfl.ch/course/view.php?id=523

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=523

Videos

- http://moodle.epfl.ch/course/view.php?id=523

EPFL

CS-410 Technology ventures in IC

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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	

Remark

Pas donné en 2021-22

Summary

This hands-on class gives graduate students in IC interested in startups the opportunity to learn and put in practice the fundamental skills required to assess a technology concept in the context of a business opportunity. This class is focused only on business opportunities where high-technology

Content

Working in teams, students will learn the fundamentals of:

- *Opportunity assessement*
- *Customer development and validation*
- *Business model alternatives*
- *Intellectual Property*
- *Strategy and Financial planning*
- *Go-to-market, launch, and growth*

This is a hands-on class where students start the class with their own technology venture concept (e.g. the work done as part of their PhD, or some well-formed idea, maybe with a prototype). During the class, they convert their concept into a integrated business plan.

Keywords

Entrepreneurship, startups, technology transfer, intellectual property

Learning Prerequisites

Required courses

- *None – but available to MS and Ph.D. students only*

Learning Outcomes

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

- Analyze a business plan
- Create a business plan

Teaching methods

- Short ex-cathedra presentations of each topic
- Hands-on seminar with many short student presentations
- Presentations from invited guests, in particluar industry executives and entrepeneurs
- Discussion and case studies

Assessment methods

- In-class participation (30%)
- In-class presentations (30%)
- Final pitch (40%)

Supervision

Office hours	Yes
Assistants	No
Forum	Yes

EPFL

CS-458 **The GC maker project**

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
SC master EPFL	MA2, MA4	Opt.

Language	English
Credits	6
Session	Summer
Semester	Spring
Exam	During the semester
Workload	180h
Weeks	14
Hours	**6 weekly**
Practical work	6 weekly
Number of positions	

Remark

pas donné en 2021-22

Summary

The GC Maker Project is an interdisciplinary project course where students work in teams towards solving real-world challenges by leveraging geometric computing methods and digital fabrication technologies.

Content

At the beginning of the course we will identify 3-4 interdisciplinary teams with complementary skills and expertise. Each team will work on a specific computational design challenge chosen by the team members in consultation with the teachers. The main focus will be on topics that combine geometry, computing, engineering, and digital fabrication to achieve the project goals. We will follow a design thinking methodology and develop a suitable project plan for each team.
Geometric and algorithmic foundations and implementations will be discussed on demand when identified during project development as necessary to achieve specific project goals.
Students will have access to a variety of digital fabrication machines, such as laser cutters, CNC milling machines, or 3D printers, and will receive appropriate training to explore different prototyping options. This will enable a cycle of ideation, code development, rapid prototyping and evaluation to progressively solve the chosen design challenge. We will define a suitable format to present project outcomes in a public forum in the final week of the course.

Learning Prerequisites

Recommended courses

CS-457 Geometric Computing is highly recommended

Important concepts to start the course

This course is a project course with limited capacity for 20 students. Students will be selected based on a 1-page letter that needs to be submitted before the semester starts describing their background and laying out their motivation for taking the course.

Learning Outcomes

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

- Apply a design thinking methodology in a computational fabrication project
- Evaluate how to best integrate computational methods and digital fabrication tools to achieve project goals
- Develop and implement geometric computing algorithms relevant for the project goals
- Assess own project progress and devise adaptations of the project plan if necessary

- Provide constructive feedback on other groups' projects
- Communicate effectively with collaborators from different disciplines
- Design a suitable format and material for public presentation of project outcomes

Transversal skills

- Assess progress against the plan, and adapt the plan as appropriate.
- 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.

Teaching methods

- Tutoring throughout the design cycle
- Hands-on tutorials on digital fabrication technologies
- Regular project critiques
- Interspersed lectures to deep-dive into specific topics, such as theoretical concepts, algorithmic foundations, engineering background, digital fabrication technologies

Expected student activities

- Coordinate project team and engage in collaborative problem solving
- Implement/adapt geometric computing algorithms
- Fabricate and evaluate prototypes
- Discuss project progress in class
- Provide constructive criticism and feedback to other groups
- Present project outcome in a public forum

Assessment methods

project assement throught the semester, final presentation

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	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	3 weekly
Exercises	1 weekly
Number of positions	

Remark

Cours biennal - pas donné en 2021-22

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

EPFL

CS-444

Virtual reality

Boulic Ronan

Cursus	Sem.	Type
Computer science	MA2, MA4	Opt.
Cybersecurity	MA2, MA4	Opt.
Data Science	MA2, MA4	Opt.
Digital Humanities	MA2, MA4	Opt.
Learning Sciences		Obl.
Robotics, Control and Intelligent Systems		Opt.
SC master EPFL	MA2, MA4	Opt.

Language	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

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 techniques for achieving efficient VR applications.

Content

The first lectures focus more on the technical means (hw & sw) for acheiving the hands-on sessions:

- Visual display
- Interaction devices and sensors
- Software environment (UNITY3D)

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 ?
- Motion capture for full-body interaction
- VR, cognitive science and true experimental design

Keywords

3D interaction, display, sensors, immersion, presence

Learning Prerequisites

Required courses

(CS 341) Introduction to Computer Graphics

Recommended courses

(CS 211) Introduction to Visual Computing

Important concepts to start the course

from Computer Graphics:
- perspective transformations
- representation of 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

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. 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.

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 quizzes and final oral

Assessment methods

Throughout semester: 1 paper citation study (20%), 1 project (50%), 1 theoretical oral (30%)

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
- J. Jerald, The VR Book, ACM Press 2015
- Parisi, Learning Virtual Reality, O’Reilly 2015
- Le Traité de Réalité Virtuelle (5 vol.) Presses des Mines, ParisTech, 2006-2009, available on-line, free for student upon registration.

- Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Ressources en bibliothèque

- 3D User Interfaces: Theory and Practice / Bowman
- Le Traité de Réalité Virtuelle / Fuchs
- The VR Book / Jerald
- Learning Virtual Reality / Parisi

Notes/Handbook

pdf of slides are made visible after the ex-cathedra courses

Websites

- http://www.thevrbook.net/

Moodle Link

- http://moodle.epfl.ch/course/view.php?id=6841

CS-503

Visual intelligence : machines and minds

Zamir Amir

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Data Science	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language	English
Credits	5
Session	Winter
Semester	Fall
Exam	During the semester
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 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. The primary focus of the course will be on embodied intelligence and perception 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 inspirations around what the missing capabilities are and how to approach them, we will turn to visual perception in biological organisms.
The course has a heavy emphasis on projects and hands-on experience. The course project will be around designing, implementing, and testing a solution to an open problem pertinent to visual perception. The students are encouraged to work in groups, self-propose a project that makes them excited, 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 and executing their course projects. In particular, the lectures will discuss:
1) A recap of basic computer vision concepts: classification, detection, segmentation, transformations, optical flow, 3D from X, etc, 2), What/why/how of visual representations. Supervised, self-supervised, unsupervised learning of representations. 3), Psychology of the visual system. 4), Physiology of the visual system. 5), Perception-action loop: active perception and embodied intelligence.
The course is of interest to MS/PhD students interested in research in computer vision, machine learning, and perceptual robotics as well as senior undergraduate students interested in gaining an advanced understanding of SOTA computer vision.

Keywords

Computer vision, machine learning, cognition, embodied intelligence, robotics, neural networks, AI.

Learning Prerequisites

Required courses

Introduction to Machine Learning (CS-233) or Machine Learning (CS-433) or equivalent course on the basics of machine learning and deep learning.

Recommended courses

Computer vision (CS-442) or equivalent undergraduate course on the basics of computer vision.

Important concepts to start the course

• Python programming.
• Basics 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:

• 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 and demonstrate continuous progress throughout the semester.

Assessment methods

• Project (70%) [Project proposal, Project checkpoint reports, Final project report and presentation]
• Homeworks (20%)
• Class attendance and engagement (10%)

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

Ressources en bibliothèque

- Vision science : photons to phenomenology / Palmer
- The Ecological Approach to Visual Perception /Gibson
- Computer Vision: Algorithms and Applications / Szeliski (2022? ; online drafts)

Notes/Handbook

The reference reading of different lectures will be from different books (main ones listed abow) and occasionally from papers. Resources will be provided in class. Full-text books are not mandatory.