

---

**Plan of studies**  
**3-5-8**

---

**2021/22**

**2021 /**  
**22**



---

# **Plan of studies**

**3-5-8**

**2021/22**

---

# Index

---

<b>Preface</b>		7	
<hr/>			
<b>Academic calendar</b>		11	
<hr/>			
<b>Faculty of Informatics</b>	Faculty of Informatics	15	
	Executive bodies	18	
	Faculty's governing bodies	21	
	Research institutes	23	
	Partner institutes	26	
	Rector, Administration and Services	29	
<hr/>			
<b>Bachelor</b>	Bachelor degree programme	35	
	Study plan	39	
	Course descriptions	41	
		First year	41
		Second year	53
		Third year	65
<hr/>			
<b>Master</b>	Informatics	79	
	Artificial Intelligence	85	
	Computational Science	88	
	Financial Technology & Computing	91	
	Management & Informatics	94	
	Software & Data Engineering	98	
	Master per l'insegnamento dell'informatica	101	
<hr/>			
<b>PhD programme</b>		103	
<hr/>			
<b>Lecturers' profiles</b>		107	

---

## Preface

Informatics is everywhere. Its impact on our everyday lives is already profound, and its reach will only deepen in the future. It is a discipline that is at the heart of today's information society, made possible by over half a century of rigorous research. Yet it also faces incredible challenges: How can we ensure that airplanes and nuclear reactors are not failing due to errors in their programming? How can we protect critical infrastructure such as hospitals and water supply plants from hackers? How do we spot and remove bias in automated systems such as credit scoring and predictive policing? And will recent advances in artificial intelligence ultimately lead to a "super-intelligence" that will eventually take over the world?

All this is ultimately governed by countless ones and zeros, the "bits" that instructs computers what to do. Informatics is the science of representing, organizing, storing, accessing, communicating, and processing these bits – i.e., information – using a universal language and a powerful formalism to describe and analyse problems and solutions. It brings together mathematics, physics, engineering, but also psychology, sociology and design, in order to describe and automate processes with the help of computers. Informatics is a mix of imagination and skills, and also a sense of beauty. And it is absolutely fascinating!

Informatics is our passion. The Faculty of Informatics is home to a diverse group of excellent researchers and dedicated teachers. It is ranked consistently among the top computer science departments in Europe and – according to a recent study by Stanford University – features over a dozen "World's Top-2%" scientists. Faculty members are engaged in several national and international research projects, and we offer a full curriculum that includes Bachelor, Master, and PhD programmes, all taught in English. The Faculty strives to offer an enthusiastic, exciting, and vibrant environment for students and researchers, and we very much look forward to welcoming you!

Prof. Marc Langheinrich  
Dean of the Faculty of Informatics

---

## Academic calendar

The academic year goes from September until June. Courses are held from September until December and from February until June. The semester includes 3 exam sessions (January, June and September).

<b>Autumn semester 2021–22</b>	Matriculation	06-24.09.2021
	Courses start	20.09.2021
	Holidays All Saints' Day • 01.11.2021 Immaculate Conception • 08.12.2021	
	Courses end	24.12.2021
<b>Exam sessions</b>	Winter session Bachelor and Master	10.01-04.02.2022
<b>Spring semester 2022</b>	Courses start	21.02.2022
	Easter holidays	15-23.04.2022
	Dies academicus	07.05.2022
	Holidays Ascension Day	26.05.2022
	Courses end	03.06.2022
<b>Exam sessions</b>	Summer session Bachelor and Master	07.06-01.07.2022
	Autumn session Bachelor and Master	29.08-16.09.2022

---

**Faculty of Informatics**

Established in October 2004, USI's Faculty of Informatics is dedicated to high quality teaching and research. The mission of the Faculty is to conduct research and produce results in the field of informatics and to equip students with creative problem-solving skills that enable them to address complex problems in business and society.

The Faculty features eight main areas of research, namely: Software Engineering, Computer Systems, Computational Science, Geometric and Visual Computing, Information Systems, Intelligent Systems, Programming Languages, and Theory and Algorithms. The Faculty is host to five institutes: the Institute of Computing (CI), the Software Institute (SI), the Computer Systems Institute (SYS), the Euler Institute (EUL) and the Dalle Molle Institute for Artificial Intelligence (IDSIA). It is also home to the affiliated Institute of Solar Physics Research (IRSOL) in Locarno.

Teaching excellence is assured by an international faculty, low student/academic staff ratio and a modern, innovative curriculum. The tuition language in the Faculty is English. The undergraduate programme is project-based and comprises six semesters of highly integrated courses and team-oriented projects. For graduate students, the Faculty offers several specialized Master's programmes (two of which in cooperation with the Faculty of Economics) and a research-oriented PhD programme. The PhD programme is highly selective and gives students the opportunity to participate in national and international research projects.

The Faculty has an active network of research partnerships with other Swiss and international centres. The national and international networks support research collaborations and student mobility.

## Executive bodies

### Dean

Prof. Marc Langheinrich  
tel +41 58 666 4304  
e-mail dean.inf@usi.ch

### Vice-Dean for Education

Prof. Patrick Eugster  
tel +41 58 666 4913  
e-mail patrick.eugster@usi.ch

### Vice-Dean for Strategy & Development

Prof. Ernst Wit  
+41 58 666 4952  
ernst.jan.camiel.wit@usi.ch

The Dean and Vice-Deans  
are available for meetings by appointment.

### Dean's Office

Elisa Larghi,  
Sabina Brambilla,  
Nadia Ruggiero,  
Jacinta Vignini  
office East Campus, Office D0.07  
tel +41 58 666 4690  
fax +41 58 666 4536  
e-mail decanato.inf@usi.ch  
bookings.inf@usi.ch  
Office hours 09.15-12.30

### Coordinator of Faculty activities/projects and external relations:

Ing. Mauro Prevostini  
tel +41 58 666 4717  
e-mail mauro.prevostini@usi.ch

### Bachelor in Informatics

Prof. Laura Pozzi  
tel +41 58 666 4301  
e-mail bachelor.dir.inf@usi.ch

### Master in Informatics

Prof. Kai Hormann  
tel +41 58 666 4327  
e-mail kai.hormann@usi.ch

Prof. Evanthia Papadopoulou  
tel +41 58 666 4122  
e-mail evanthia.papadopoulou@usi.ch

### Master in Artificial Intelligence

Prof. Luca Maria Gambardella  
tel +41 58 666 4317  
e-mail luca.maria.gambardella@usi.ch

Prof. Jürgen Schmidhuber  
tel +41 58 666 4318  
e-mail jurgen.schmidhuber@usi.ch

### Master in Computational Science

Prof. Olaf Schenk  
tel +41 58 666 4850  
e-mail olaf.schenk@usi.ch

Prof. Ernst Wit  
tel +41 58 666 4121  
e-mail ernst.jan.camiel.wit@usi.ch

### Master in Financial Technology & Computing

Prof. Marc Langheinrich  
tel +41 58 666 4304  
e-mail marc.langheinrich@usi.ch

Prof. Fernando Pedone  
tel +41 58 666 4695  
e-mail fernando.pedone@usi.ch

### Master in Management & Informatics

Prof. Marc Langheinrich  
tel +41 58 666 4304  
e-mail marc.langheinrich@usi.ch

**Master in Software & Data Engineering**

Prof. Cesare Pautasso

tel +41 58 666 4311  
 e-mail cesare.pautasso@usi.ch

Prof. Gabriele Bavota

tel +41 58 666 4928  
 e-mail gabriele.bavota@usi.ch

**PhD programme Director**

Prof. Walter Binder

tel +41 58 666 4303  
 e-mail phddirector.inf@usi.ch

Prof. Stefan Wolf

tel +41 58 666 4121  
 e-mail phddirector.inf@usi.ch

**Delegate for International Relations**

Prof. Paolo Tonella

tel +41 58 666 4848  
 e-mail paolo.tonella@usi.ch

The programme directors and delegates  
 are available by appointment.

**Faculty's governing bodies**

The Faculty's governing bodies include: the Faculty Council, the Professors Council, and the Dean's Office.

**Faculty Council**

The highest body of the Faculty is the Faculty Council. It comprises:

- all tenured professors (full and associate), the assistant professors and adjunct professors of the Faculty;
- one teacher representative (with one- or two-year contract),
- one post-doctoral researcher representative, one PhD student representative and one student representative (Bachelor and Master)

**Full professors**

Cesare Alippi  
 Walter Binder  
 Michael Bronstein  
 Antonio Carzaniga  
 Fabio Crestani  
 Patrick Eugster  
 Luca Maria Gambardella  
 Illia Horenko  
 Kai Hormann  
 Rolf Krause  
 Marc Langheinrich  
 Michele Lanza  
 Evanthia Papadopoulou  
 Cesare Pautasso  
 Fernando Pedone  
 Mauro Pezzè  
 Igor Pivkin

Laura Pozzi  
 Olaf Schenk  
 Jürgen Schmidhuber  
 Natasha Sharygina  
 Paolo Tonella  
 Ernst Wit  
 Stefan Wolf

<b>Associate professors</b>	Gabriele Bavota Carlo A. Furia Matthias Hauswirth Silvia Santini
<b>Assistant professors</b>	Piotr Didyk Michael Multerer
<b>Adjunct professors</b>	Andrea-Emilio Rizzoli Robert Soulé
<b>Faculty Representatives</b>	<ul style="list-style-type: none"> <li>• Students Michele Cattaneo (Roberto Palmieri)</li> <li>• PhD Souhaila Serbout (Dylan Ashley)</li> <li>• Post-docs Ilaria Scarabottolo (Andrea Rosà)</li> <li>• Teachers Marco Brambilla</li> </ul>
<b>Professors Council</b>	The Professors Council is made up of all tenured professors (full and associate) of the Faculty.
<b>Dean</b>	The current Dean is Prof. Marc Langheinrich
	For the specific duties of each body please refer to the Statute of the Faculty.

## Research institutes

### CI Institute of Computing

Human thinking and decision making are transformed by unprecedented advances in mathematics, software and computer hardware. Computing has revolutionized our intellectual capacity to tackle complex problems and it is transforming the arts, sciences and every aspect of our society. Computing assists our commute, the storage and processing of our Data and it is instrumental in our efforts to analyze the Human Genome, to project Climate change and analyze the evolution of our Society. Computing is part of our everyday life and it has become indispensable for our efforts to overcome the complexity of our Information age. Simply put: Computing is essential for human evolution and survival.

The Institute of Computing at USI fosters a synergistic advancement of various components of computing. It will co-evolve multiple computing modes and advance mathematics, algorithms and software to address the challenges presented by scientific and societal problems. There is urgency across disciplines for developing and harnessing such new modes of computing. We envision that we will provide the foundations of this new form of computational thinking and, at the same, time tools to assist in projecting simulations and data science and its effects on humanity as well as extracting structure from massive amounts of Data as they emerge across our Society.

### Director of CI

- Prof. Ernst Wit  
<https://www.ci.inf.usi.ch/>

### SI Software Institute

The Software Institute (SI) is part of the Faculty of informatics of the Università della Svizzera italiana (USI), located in beautiful Lugano, in Southern Switzerland. At the SI, our strength is discovering, designing, and developing new ideas that ease the conception of modern software systems. Our research is rooted both in sound theoretical models as well as practical, real-life questions that impact modern society, a society where reliable, well engineered software systems have become quintessential. The SI is a center of excellence committed to the teaching, the research and the

**Director of  
SI**

development of software. The SI is directed by Michele Lanza and features renowned software researchers among its members: Profs. Gabriele Bavota (Software Analytics & Empirical Software Engineering), Matthias Hauswirth (Software Performance), Cesare Pautasso (Software Architecture & Web Engineering), Carlo Alberto Furia (Software Engineering, Formal Methods & Verification), and Paolo Tonella (Software Testing).

- Prof. Michele Lanza  
[www.si.usi.ch](http://www.si.usi.ch)

**SYS  
Computer  
Systems  
Institute**

Computer systems (“systems”) encompass all areas of computer science directly related to (or having an impact on) the design, architecture, development, deployment, and operation of software and hardware systems. Topics of interest include, e.g., operating systems, networking, distributed systems, security and privacy, real-time systems, cloud computing, data management, programming languages, middleware, ubiquitous computing, embedded systems, computer architecture, and a wide range of applications. Historically, these areas have existed independently, but the increased complexity of computing artifacts increasingly requires collaborative efforts from multiple points of view to address relevant problems.

The primary goal of the Computer Systems Institute (SYS) is to develop and promote world-class research and teaching in the area of systems. Institute members have a strong presence in the community (e.g., steering and program committees of prestigious conferences, editorial boards, and collaborations with major companies) and actively participate in major national and international research efforts (e.g., SNSF, InnoSuisse, EU, industry-sponsored initiatives). The institute also plays a major role in the Bachelor and Master educational programs (e.g., teaching roughly 30% of the core courses in the Bachelor curriculum of the Faculty of Informatics, and offering two interdisciplinary Master programs in collaboration with the Faculty of Economics).

**Director of  
SYS**

- Prof. Fernando Pedone  
<https://sys.inf.usi.ch/>

**EUL  
Euler Institute**

Human thinking and decision making are transformed by unprecedented advances in mathematics, software and computer hardware. Computing has revolutionized our intellectual capacity to tackle complex problems and it is transforming the arts, sciences and every aspect of our society. Computing assists our commute, the storage and processing of our Data and it is instrumental in our efforts to analyze the Human Genome, to project Climate change and analyze the evolution of our Society. Computing is part of our everyday life and it has become indispensable for our efforts to overcome the complexity of our Information age. Simply put: Computing is essential for human evolution and survival.

The Institute of Computing at USI fosters a synergistic advancement of various components of computing. It will co-evolve multiple computing modes and advance mathematics, algorithms and software to address the challenges presented by scientific and societal problems. There is urgency across disciplines for developing and harnessing such new modes of computing. We envision that we will provide the foundations of this new form of computational thinking and, at the same, time tools to assist in projecting simulations and data science and its effects on humanity as well as extracting structure from massive amounts of Data as they emerge across our Society.

**Director of  
EUL**

- Prof. Rolf Krause  
<https://www.euler.usi.ch/en>

**IDSIA  
Istituto Dalle  
Molle di Studi  
sull'Intelligenza  
Artificiale**  
Joint  
USI-SUPSI  
institute

IDSIA was founded in Lugano in 1988 by Angelo Dalle Molle (1908-2002), an Italian philanthropist whose vision was a world where technological progress and human development could both contribute to the improvement of our quality of life. Dalle Molle was a precursor of electric mobility, and he established a Trustee in Switzerland to promote creative scientific research, free from the bureaucratic ties of university institutions. Nowadays the institutes founded by Angelo (IDSIA in Lugano, IDIAP in Martigny, and ISSCo in Geneva) are integrated in the local institutions. Since the foundation of USI and SUPSI in Canton Ticino, IDSIA has been designated to be a “bridge” between these two institutions. For this reason IDSIA activities span from fundamental to applied research, transferring its knowledge into applications in the real world.

**Director of  
IDSIA**

- Prof. Andrea-Emilio Rizzoli  
[www.idsia.ch](http://www.idsia.ch)

**CSCS  
Swiss National  
Supercomputing  
Centre**

Founded in 1991, CSCS develops and provides the key supercomputing capabilities required to solve challenging problems in science and/or society. The centre enables world-class research with a scientific user lab that is available to domestic and international researchers through a transparent, peer-reviewed allocation process. CSCS's resources are open to academia, and are available as well to users from industry and the business sector. The centre is operated by ETH Zurich and is located in Lugano. CSCS and the Università della Svizzera italiana coordinate the Swiss Platform for Advanced Scientific Computing (PASC); a joint effort of all Swiss universities to create a long-term research-driven cooperation network in computational science. The PASC overarching goal is to position Swiss computational sciences in the emerging exascale-era and aims to provide the Swiss scientific community with the tools to make the best use of the new generations of supercomputing machines to solve key problems for science and society. It addresses important scientific research issues in high-performance computing and computational science in different domain sciences through interdisciplinary collaborations between domain scientists, computational scientists, software developers, computing centres and hardware developers.

**Director of  
CSCS**

- Prof. Thomas C. Schulthess  
info@cscs.ch  
www.cscs.ch

**Rectorate**

The Rectorate ensures the overall smooth functioning of the University and elaborates planning and development acts.

**Rector**

Prof. Boas Erez

**Secretary General**

Dr. Giovanni Zavaritt

**Administrative Director**

Cristina Largader

**Adjunct Administrative Director**

Antoine Turner

**Pro-Rector  
for Internationalisation**

Prof. Cesare Alippi

**Pro-Rector  
for Education and Students' Experience**

Prof. Lorenzo Cantoni

**Pro-Rector  
for Research**

Prof. Patrick Gagliardini

**Pro-Rector for  
Innovation and Corporate relations**

Prof. Luca Maria Gambardella

**Pro-Rector  
for Research in the Humanities and Equal Opportunities**

Prof. Sonja Hildebrand

30 **Administration and Services Lugano Campus** USI administration comprises of different services and it is under the Rectorate responsibility, and through the Rectorate, under the University Council.

#### **Alumni**

office Main Building, Office 303  
tel +41 58 666 4606  
e-mail alumni@usi.ch  
web www.usi.ch/en/alumni

#### **Career Service**

office Main Building, Office 303  
tel +41 58 666 4606  
e-mail careerservice@usi.ch  
web www.usi.ch/en/career

#### **Copy Center Onys**

office Main Building, Office 139  
tel +41 58 666 4586  
e-mail copycenter@usi.ch

#### **Corporate identity**

office Main Building, Office 212  
tel +41 58 666 4761  
e-mail corporate-design@usi.ch

#### **Development and Institutional Relations**

office Main Building, Office 232  
tel +41 58 666 4927  
e-mail maria.cristina.reinhart@usi.ch

#### **eLearning Lab (eLab)**

office Main Building, Offices 403-406  
tel +41 58 666 4760  
e-mail info@elearninglab.org  
web www.elearninglab.org/en/

#### **Equal Opportunities**

office Main Building, Office 202  
tel +41 58 666 46 12  
e-mail equality@usi.ch  
web www.usi.ch/equality

#### **Financial Controlling**

office Main Building, Office 221-222  
tel +41 58 666 4518

#### **Front Office**

office Main Building, Office 220  
tel +41 58 666 4000  
e-mail info@usi.ch

#### **Graphic design**

office Main Building, Office 210  
tel +41 58 666 4456  
e-mail grafica@usi.ch

#### **Housing**

office Main Building, Office 220  
tel +41 58 666 4489  
e-mail alloggi@usi.ch  
web www.usi.ch/en/alloggi

#### **Human resources**

office Main Building, Office 233-235  
e-mail personale.lu@usi.ch

#### **Institutional Communication Service**

office Main Building, Office 212  
tel +41 58 666 4792  
e-mail press@usi.ch

#### **L'ideatorio**

office Villa Saroli, Viale S. Franscini 9  
tel +41 58 666 4520  
web www.ideatorio.usi.ch/

#### **International Relations and Study Abroad Service**

office Main Building, Office 332  
tel +41 58 666 4626  
e-mail relint@usi.ch  
web www.usi.ch/en/relint

#### **IT**

office Main Building, Office 164  
tel +41 58 666 4610  
e-mail itsupport.lu@usi.ch  
web http://usi.4me.com/

#### **Legal**

office Main Building, Office 228  
tel +41 58 666 4616  
e-mail serviziogiuridico@usi.ch

#### **University Library Lugano**

tel +41 58 666 4509  
e-mail library.lu@usi.ch  
web www.bul.sbu.usi.ch

#### **Psychological Counselling Service**

office info@psicologi-ticino.ch  
e-mail www.psicologi-ticino.ch

**Quality Assurance**

office Main Building, Office 227  
 tel +41 58 666 4199  
 e-mail quality@usi.ch

**Research and Transfer Service**

office Main Building, Office 216/228  
 tel 41 58 666 4614  
 e-mail +servizio.ricerca@usi.ch  
 web www.ticinoricerca.ch

**Sport Service**

office East Campus, Office D0.07  
 tel +41 58 666 4797  
 e-mail sport@usi.ch  
 www.sport.usi.ch

**Study Advisory and Promotion Service**

office Main Building, Office 303 (Level 3)  
 tel +41 58 666 4795  
 e-mail studyadvisory@usi.ch

**USI Startup Centre**

office East Campus, Sector C  
 tel +41 58 666 41 40  
 e-mail startup@usi.ch  
 web www.startup.usi.ch

**Web**

office Main Building, Office 210  
 tel +41 58 666 4515  
 web web@usi.ch

**USI online services and resources****Student platform**

- [www.usilu.net](http://www.usilu.net)

**Faculty course registration platform**

- <http://teaching.inf.usi.ch>

**Faculty PhD platform**

- <https://phdprogram.inf.usi.ch>

**eCourses platform (Moodle)**

- [www.icorsi.ch](http://www.icorsi.ch)

**Exam registration and consultation:**

- [www.esami.lu.usi.ch](http://www.esami.lu.usi.ch)

**desk.usi - Practicalities for the USI community:**

- [www.desk.usi.ch](http://www.desk.usi.ch)

**Student Corporation**

The Student Corporation is the organised structure of the University student body. This means that the sole fact that a Bachelor's or Master's student is enrolled at USI makes him/her a member of an umbrella organisation that is interconnected with the institutional structure of the University. The Corporation can promote activities and projects to improve students' experience on all campuses.

The Student Corporation aims to contribute to a positive and enriching student experience by strengthening the sense of community. It gives voice to the student body and gets students involved. Its name refers to the very origins of universities that were born from the union of "corporations" (called universitates) of professors and students. The Student Corporation is non-profit, non-confessional and non-political. It is inspired by the principles of democracy and equality of all its members.

More information:

[www.usi.ch/en/university/organisation/student-corporation](http://www.usi.ch/en/university/organisation/student-corporation)

**Student associations**

Several student associations have been created within the University. The main objectives are to improve relations between students and the institution and to enrich the range of educational and recreational offer during school. The associations are concerned mainly with the collection of didactic material, organisation of parties and meetings, cultural and sporting events, and networking among University students and the business world.

[www.desk.usi.ch/en/list-acknowledged-student-associations](http://www.desk.usi.ch/en/list-acknowledged-student-associations)

More information:

[www.usi.ch/en/administration-and-services](http://www.usi.ch/en/administration-and-services)

---

**Bachelor in informatics**

**BSc**

<b>Introduction</b>	The Bachelor of Science in Informatics introduces students to the theory and practice of informatics. It emphasizes theoretical foundations, technology, systems thinking, and soft skills like communication and teamwork. The curriculum is structured around four areas of learning essential for a truly interdisciplinary education:
<b>Theory</b>	The principles and foundations were established in the 20th century. These foundations help the students understand the potential and limits of computing. The theoretical subjects represent a solid basis to conduct sound scientific analysis and design.
<b>Technology</b>	Informatics is in permanent and fast-paced evolution, characterized by rapid changes in technology. Students are exposed to the most recent technological advances and learn to cope with technological change and evolution, as well as the impact of technology on society.
<b>Systems thinking</b>	Informatics systems today form the foundations of many societal, governmental, and business systems and services. Students learn to view a computer-based system as a component of a larger environment rather than an isolated system. Communication and Teamwork. Information technology projects are intrinsically interdisciplinary. Informatics professionals work in teams to identify complex problems and develop appropriate solutions. Students learn to communicate, to work with others in teams, and to present the results of their work. The program is based on the European Credit Transfer System (ECTS), which is recognized by all universities in Europe. The three-year Bachelor degree (BSc) is followed by a two-year graduate study programme, leading to a Master degree (MSc). The Faculty offers six Master programmes (see p. 79).
<b>Mobility</b>	A student can take part in a mobility or student exchange programme and undertake a semester in another university for a maximum of 30 ECTS in one semester. The student must discuss the choice of host institution and the study plan with the Bachelor

director and obtain approval. The mobility period generally lasts one semester; it may be extended, subject to approval of the Bachelor director, to a maximum of two consecutive semesters. For all information about mobility please consults the International Relations and Study-abroad Office at <http://www.usi.ch/en/relint>.

### Study plan

The Bachelor programme consists of an innovative, project-based, team-oriented curriculum of six semesters (three years) and corresponds to 180 ECTS credits proportionally distributed (30 ECTS for each semester). In the first four semesters, students work on group projects. In the fifth semester students are required to do an internship (field project) in industry. In the sixth semester, they work on an individual final project in which they use all the acquired knowledge to solve an interesting problem. The Bachelor students have opportunities for summer internships both at companies and at the university.

## Study programme Bachelor curriculum

	Course	Instructor	ETCS
<b>First semester</b> 30 ETCS	Calculus	Hormann	6
	Computer Architecture	Langheinrich	6
	Programming Fundamentals 1	Furia	9
	Reason and Responsibility in Decision Making	Carzaniga	3
	Software Atelier 1: Fundamentals of Informatics	Bavota	6
<b>Second semester</b> 30 ETCS	Algorithms & Data Structures	Carzaniga	6
	Discrete Structures	Wolf	6
	Linear Algebra	Pivkin	6
	Programming Fundamentals 2	Hauswirth	6
	Software Atelier 2: Human-Computer Interaction	Santini	6
<b>Third semester</b> 30 ETCS	Automata & Formal Languages	Pozzi	3
	Probability & Statistics	Wit	6
	Programming Fundamentals 3	Binder	6
	Systems Programming	Carzaniga	6
	Software Atelier 3: The Web	Pautasso	9
<b>Fourth semester</b> 30 ETCS	Computer Networking	Santini	6
	Data Management	Eugster	6
	Introduction to Computational Science	Multerer	3
	Operating Systems	Pedone	6
	Software Atelier 4: Software Engineering Project	Mocci	9
<b>Fifth semester</b> 30 ETCS	Algorithms & Data Structures 2	Papadopoulou	3
	Artificial Intelligence	Gambardella	3
	Computer Graphics	Didyk	6
	Information Retrieval	Crestani	6
	Numerical Computing	Schenk	6
	Software Atelier 5: Field Project	Lanza, Prevostini	9
<b>Sixth semester</b> 30 ETCS	Experimentation & Evaluation	Bavota	3
	Machine Learning	Alippi	6
	Optimization Methods	Krause	6
	Theory of Computation	Sharygina	6
	Bachelor Project	Pezzè	12

---

**Bachelor of Science**

---

**First year**

---

---

Lecture  
6 ECTS

---

## Calculus

---

**Instructor** Kai Hormann

---

**Course Objectives** This course teaches the essentials from real analysis, which are relevant to informatics. Whenever possible, applications of theoretical concepts are highlighted. After finishing this course, students possess the mathematical skills required for solving basic problems in a formal and structured way and they will have developed a good understanding of differential and integral calculus

---

**Course Description** This course consists of five chapters. After revisiting basic facts about natural numbers, integers, and rational numbers, the first milestone is to understand the concept of real numbers and their properties. We then study sequences and series of real numbers and learn about the idea of convergence. The third chapter introduces real functions in one variable and focuses on the property of continuity and its consequences. Differentiation and integration are covered in the last two chapters.

---

**Learning Methods** The topics are presented in the form of lectures and tutorials. Weekly homework assignments are handed out, graded, and discussed.

---

**Examination Information** The course grade is determined by the results of the homework assignments (30%), the written midterm exam (30%), and the written final exam (40%).

---

**References**

- Kenneth A. Ross. *Elementary Analysis: The Theory of Calculus*. Undergraduate Texts in Mathematics. Second Edition. Springer, 2012  
Additional material will be provided through the course homepage.

Lecture  
6 ECTS

## Computer Architecture

**Instructor** Marc Langheinrich

**Course Objectives** After successfully completing the course, students will know the basic principles of how a computer functions, from the very basic building blocks (transistors and logical gates) to the more complex components (CPU, memory, buses, I/O interfaces).

**Course Description** In this course, students learn how one can describe the basic operations in a computer using digital logic, and how these operations can be realized in both hardware and software. Students gradually combine these basic operations into a "microarchitecture" -- a software-controlled datapath that connects digital memory with an arithmetic-logical unit -- on which one can then build more and more complex "layers" that will finally allow the writing of complex programs in human-readable programming languages. This knowledge not only forms the basis for understanding how something as complex as a modern computer actually works, but is also a pre-requisite for learning about many advanced topics in informatics.

**Learning Methods** Frontal lectures, weekly assignments, lab sessions

**Examination Information** The final grade will be based on a student's Midterm (30%) and Final (70%) exam scores. Optional assignments and/or quizzes (corrected but ungraded) will offer continuous learning feedback throughout the semester.

**References**

- *Structured Computer Organisation*, Andrew S. Tanenbaum, Todd Austin. 6th Edition (International), Pearson Education, 2012, ISBN-10: 0273769243, ISBN-13: 978-0273769248

Lecture & Lab  
9 ECTS

## Programming Fundamentals 1

**Instructor** Carlo A. Furia

**Course Objectives** This is a first programming course that teaches you how to program well. "Programming well" requires more than writing code until it seems to work. This course emphasizes designing programs: constructing software in a principled manner, following a process that goes from problem statement to implementation and is based on justifiable choices about abstractions, data structures, algorithms, and program organization.

**Course Description** The course uses the programming language Racket, but it is not a course about Racket, which we use primarily because of the conceptual clarity of its features. Once you master this course's content, learning another programming language won't be a problem. The course is suitable for students with little or no prior programming experience, but even programmers with experience will have a chance to learn new program design principles and to practice on interesting problems.

**Learning Methods** The course's organization includes plenty of opportunities to practice programming — including assignments, exercises, and a final project.

**Examination Information** The course's grades is determined by a combination of programming assignments developed during the course, a final course project, and a final exam.

**References**

- M. Felleisen, R. B. Findler, M. Flatt, S. Krishnamurthi: *How to Design Programs*, Second Edition, 2014. <https://htdp.org/2020-5-6/Book/>

Lecture  
3 ECTS

## Reason and Responsibility in Decision Making

**Instructor** Antonio Carzaniga

**Course Objectives** In life we face all sorts of decisions. Some are consequential and also often entail conflicting goals and uncertain outcomes, which makes them difficult. The main objective of this course is to understand the nature of the decisions so as to assess their consequences with the most appropriate methods.

**Course Description** This course discusses general methods to reflect on the nature of problems to then make responsible decisions. The methods range from logic and mathematics to experimental science, to personal dialogue and ethical valuation.

**Learning Methods** Interactive discussions based on examples and case studies presented by different professors with a variety of styles, methods, and competences.

**Examination Information** Participation; final essay on the analysis of one or more case-study decisions.

Atelier  
6 ECTS

## Software Atelier 1: Fundamentals of Informatics

**Instructor** Gabriele Bavota

**Course Objectives** To become familiar with LaTeX, HTML, CSS, Versioning, and the unix shell. To learn key events in the history of computer science. To improve soft skills.

**Course Description** The first of the ateliers, which are a crucial part of our Bachelor curriculum is roughly divided into three main pieces. On the one hand the students will obtain first-hand experience with a variety of tools of the trade, such as LaTeX, HTML, Versioning, and the unix shell. Second, the students will get an overview of the history of computer science since its very beginning up to the present day. The third part of the atelier is dedicated to a group project, in which students will put into practice what they learned in the course.

**Learning Methods** Hands-on sessions in class covering the course topics, assignments, group project, presentations in class.

**Examination Information** The final grade will be defined based on the assignments submitted by the student throughout the semester.

**References** Material provided during the course.

Lecture  
6 ECTS

## Algorithms & Data Structures

**Instructor** Antonio Carzaniga

**Course Objectives** Algorithms and data structures are fundamental to computer science. They are the essence of computer programs. The performance of any software system depends on the efficiency of its algorithms and data structures, and more generally, the study of algorithms provides insights into the nature of problems. This course teaches algorithmic programming, meaning how to solve problems with computers, and how to design and reason about algorithms. The course also teaches the purpose as well as the strengths and weaknesses of some of the most fundamental algorithms and data structures.

**Course Description** The course covers basic notions of: complexity, asymptotic worst-case and average complexity, big-O notation, complexity classes; general algorithmic strategies, brute force, greedy, divide-and-conquer, and dynamic programming; common algorithms, searching and sorting, elementary graph algorithms, basic data structures, stacks, queues, linked lists, rooted trees; more advanced data structures, B-trees, heaps, hash tables.

**Learning Methods** Theoretical lectures combined with a lot of programming exercises.

**Examination Information** Homework assignments, midterm exam, final exam, all consisting of algorithmic problems and programming exercises.

**References** Textbook:  

- *"Introduction to Algorithms"* (Third Edition), by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, published by MIT Press and McGraw-Hill.

Lecture  
6 ECTS

## Discrete Structures

**Instructor** Stefan Wolf

**Course Objectives** This course's goal is to learn principles of the mathematics of countable structures. Hereby, central themes are modeling, abstraction, simplification, and generalization.

**Course Description** The main topics of the course are propositional logic and proofs; sets, relations, and functions; combinatorics (urn models, inclusion-exclusion), graph theory (trees, planar graphs, Euler tours and Hamilton cycles) and some basic number theory (modular calculus, groups, Euler's theorem, RSA).

**Learning Methods** Lectures and assignments.

**Examination Information** Midterm: 2 hrs written & Final: 3 hrs written or Final: 15 min oral

**References** Course Notes

Lecture  
6 ECTS

## Linear Algebra

**Instructor** Igor Pivkin

**Course Objectives** At the end of the course, the student will be able to use the major concepts of linear algebra.

**Course Description** This course gives an introduction to the field of linear algebra. Concepts and techniques from linear algebra are of fundamental importance in many scientific disciplines and provide the "language" for understanding the behavior of linear mappings and linear spaces. Topics covered are linear systems and Gauss method, vector spaces, linear maps and matrices, determinants, eigenvectors and eigenvalues.

**Learning Methods** The course will use lectures, tutorials and practicals.

**Examination Information** The course will be evaluated through homework, midterm and final exams.

**References**

- *"Introduction to Linear Algebra"* by Gilbert Strang
- *"Introduction to Linear Algebra"* by Serge Lang

Lecture & Lab  
9 ECTS

## Programming Fundamentals 2

**Instructor** Matthias Hauswirth

**Course Objectives** In this course you learn how to develop software using an object-oriented approach. You will learn the underlying principles and their realization in the Java programming language

**Course Description** The course teaches how to structure a problem using the concept of classes, and how to use fields and methods to model state and behavior. It introduces the fundamental concepts of types, dynamic memory allocation, and references. It covers the ideas of collections and iteration to deal with multiple objects, the ideas of inheritance and polymorphism to deal with variability in software, and the idea of exception handling to deal with unexpected situations. It covers principles of design such as coupling and cohesion, encapsulation, and immutability, and it introduces common design patterns. The practical aspects of the course include testing and debugging techniques that help improve the quality of the resulting software.

**Learning Methods** Lectures, textbook, programming assignments, mastery checks, code reviews.

**Examination Information** Written in-class final exam (if possible).

**Prerequisites** Computer Architecture, Programming Fundamentals 1, Software Atelier 1: Fundamentals of Informatics

**References**

- See *"Programming Fundamentals 2"* course web site on <http://informa.inf.usi.ch>

Atelier  
3 ECTS

## Software Atelier 2: Human-Computer Interaction

**Instructor** Silvia Santini

**Course Objectives** Goal of the course is to provide students with theoretical and practical knowledge of human-centered design of interfaces for computing systems.

**Course Description** The course starts with an introduction to ethics that investigates the many ways and venues information and communication technologies can be used maliciously. The course moves on to introduce the concept of user-centred design to produce usable, useful and used tools. The different stages from ideation to paper prototyping are covered in theory and practice.

**Learning Methods** Theoretical concepts will be provided through instructor-led lectures. Theory will be put to good use in practice, by letting students engage in group work to deliver a project, developed through progressive stages of design and prototyping.

**Examination Information** The final grade will result from the combination of the grades of the individual assignments.

**References** Textbook:

- David Benyon. *Designing User Experience: A guide to HCI, UX and interaction design*. Pearson, 4th edition, January 2019.

Further material will be provided during the semester.

## Bachelor of Science

### Second year

---

Lecture  
3 ECTS

---

## Automata & Formal Languages

---

**Instructor** Laura Pozzi

---

**Course Objectives** The theory of automata and formal languages deals with the problem of modeling computation: what is a computer, and what are its fundamental capabilities? Thus, it constitutes the basis for further studies on the theory of computability and complexity. Additionally, Automata and Formal Languages is a very practical course, as it provides knowledge of the models used in many branches of computer science, from scanners and lexical analyzers in compilers, to programs for designing digital circuits, and even in other areas such as linguistics. At the end of this course you will be familiar with models of computations used today, you will understand how they are fundamental to further studies and you will be ready for a more advanced course on the theory of computation.

---

**Course Description** Regular Languages (Finite Automata, Non Deterministic Finite Automata, Regular Expressions, Pumping Lemma for regular languages) and Context Free Languages (Context -Free Grammars and Push Down Automata).

---

**Learning Methods** Lessons, Assignments, Exams.

---

**Examination Information** The grade will be based on your performance on assignments, on a midterm exam and on a final exam.

Lecture  
6 ECTS

## Probability & Statistics

**Instructor** Ernst Wit

**Course Objectives**

- learn about probability axioms and (conditional) probability distributions
- study moments of random variables and Chebychev's inequality
- learn about law of large numbers and central limit theorem
- study estimation methods and ways to evaluate estimators
- develop hypothesis testing
- study linear regression

**Course Description**

Probability theory is a deductive science describing the axioms for calculating the probability of some event given some known state of the world. In the first part of the course, we define the probability axioms, introduce the concept of events, random variables, and probability distributions. In inductive practice we are interested to learn about the state of the world given some event, i.e., the data. Statistics is probability theory turned upside down. In this course we will learn about estimation procedures, in particular, maximum likelihood and the method of moments. We will introduce hypothesis testing and linear regression analysis.

**Learning Methods** Combination of lectures and tutorials

**Examination Information** Written exam

**References**

- *Prasanna Sahoo, Probability and Mathematical Statistics, 2013.*

Lecture  
6 ECTS

## Programming Fundamentals 3

**Instructor** Walter Binder

**Course Objectives**

- Concepts of concurrent and parallel programming.
- Parametric polymorphism.
- Functional programming.

**Course Description**

This course teaches concepts and methods of concurrent and parallel programming, with particular focus on multi-threading, safety/liveness hazards, and synchronization. The course also covers patterns, constructs and classes made available by the Java language and the Java Class Library to implement thread-safe concurrent applications. Moreover, the course introduces generic programming in Java, including related concepts such as erasure and reification. Finally, the course teaches principles of functional programming and its application on the Java Virtual Machine, including lambdas and Java streams. The presented techniques enable the development of scalable, concise, easy-to-maintain and flexible multi-threaded Java applications capable of exploiting modern multicore hardware.

**Learning Methods** Lectures and exercises on the course topics.

**Examination Information** The examination consists of periodic assignments and a final exam.

**References**

- *Java Concurrency in Practice* (2006), by B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea. ISBN-13: 978-0321349606.
- *Java Generics and Collections* (2007), by M. Naftalin and P. Wadler. ISBN-13: 978-0596527754.
- *Mastering Lambdas: Java Programming in a Multicore World* (2014), by M. Naftalin. ISBN-13: 978-0071829625. Materials made available by the course instructors.

Lecture  
6 ECTS

## Systems Programming

**Instructor** Antonio Carzaniga

**Course Objectives** A "system" integrates components and functionality at different levels. Examples are information systems consisting of databases and processing modules; a distributed storage system consisting of networked redundant storage devices; an operating system that manages heterogeneous computing resources; a robotic system made of physical devices, embedded sensors and controllers, as well as complex processing modules. The most common systems programming language is C. This course is a practice-oriented introduction to programming in C and, to a limited extent, C++. The goal is not only to learn how to write correct C programs, but also to understand their execution model in depth.

**Course Description** The course focuses on the features of the C language and libraries that are particularly useful in programming systems. This includes the memory model, input/output, modularization, its relevant language features, and its relation to the build process.

**Learning Methods** This is a practice-driven course. All topics are introduced and explained through concrete working examples.

**Examination Information** The evaluation is based on concrete programming problems, with a combination of homework assignments and exams also consisting of programming exercises.

**References** There are no specific required textbook, although students are encouraged to read *The C Programming Language* (Second Edition) by Brian W. Kernighan and Dennis M. Ritchie.

Atelier  
9 ECTS

## Software Atelier 3: The Web

**Instructor** Cesare Pautasso

**Course Objectives** The ultimate goal of the Informatics Atelier is to teach the student to become a computing professional. To this end, the atelier gives an introduction to the role of computing and computer scientists in the professional world as well as society in general and provides an environment for the students to learn about and use specific software tools, work with other students in group projects, and effectively present the results of their projects.

**Course Description** The emphasis during the Web Atelier in the third semester is on client/server programming, model-view-controller, emerging Web technologies and Web design. The Web Atelier will cover the following Web technologies: REST and HTTP, CSS3, HTML5, JSON; students will also learn how to program in JavaScript on the client and on the server-side with Node.JS, the Express framework and MongoDB.

**Learning Methods** Students will be supported during their learning with mastery checks, giving them continuous feedback as they progress through the foundational material. Once they have shown mastery, they can start working on projects.

**Examination Information** No final exam. Students will be evaluated based on mastery checks and their contribution to team projects.

**Prerequisites** Computer Networking, Data Management, Programming Fundamentals 1, Programming Fundamentals 2

**References** Handouts during the theoretical part of the atelier will complement freely available online tutorials.

Lecture  
6 ECTS

## Computer Networking

**Instructor** Silvia Santini

**Course Objectives** The Internet provides global connectivity for applications and end-users. This course aims to convey an understanding of, on the one hand, how common distributed applications such as the world-wide web use the network (the Internet), and, on the other hand, how the network is designed and how it provides its services to applications. This course serves the designers of distributed applications, as well as network designers as an introduction to computer networking

**Course Description** The course covers the architecture of networked applications and the network itself, their fundamental protocols, and the design principles behind them. This includes applications such as the Web, e-mail, and peer-to-peer systems; the two most important transport protocols of the Internet, namely UDP and TCP; the network layer; and basics of the link and physical layer.

**Learning Methods** The course includes instructor-led lectures and interactive practical sessions. Several assignments will be provided during the semester and discussed in class.

**Examination Information** The final grade will result from a combination of the grade obtained through the final exam and the grades of assignments and of the mid-term exam.

**References** Textbook:

- *“Computer Networking: A Top-Down Approach”*, by James F. Kurose and Keith W. Ross, published by Addison-Wesley.

Lecture  
6 ECTS

## Data Management

**Instructor** Patrick Eugster

**Course Objectives** This course provides a practical introduction to database technology. By the end of this course, students will understand the fundamental concepts about database management systems, become familiar with commercial tools for the design and development of database applications, and be exposed to recent trends in database-like storage systems.

**Course Description** Databases are essential to applications in a wide variety of domains, including finance, health care, commerce, and telecommunications. In fact, most applications that people use on a day-to-day bases are backed by databases. Topics covered include modeling enterprise data with ER diagrams, the relational model, SQL, XML, logical design with normalization, physical design, query execution, transaction processing, recovery, concurrency, online analytical processing, and NoSQL systems.

**Examination Information**

- Midterm, written
- Assignments
- Final exam, written

**References recommended**

- *Database Management Systems*, Ramakrishnan & Gehrke, 3rd ed.
- *Fundamentals of Database Systems*, Elmasri & Navathe, 6th ed.

Lecture  
6 ECTS

## Introduction to Computational Science

**Instructor** Michael Multerer

**Course Objectives** Knowledge and understanding of basic numerical algorithms and their implementation.

**Course Description** Computational science is an interconnected combination of computer science and mathematics in which we develop and analyse algorithms for solving important problems in science and engineering. In this course, students will learn principles and practices of basic numerical computations. This class will cover in particular: numerical solution of linear systems, interpolation and approximation, numerical quadrature, least squares and one dimensional non-linear equations.

**Learning Methods** Depending on the teaching situation in spring either direct instruction plus exercises (offline) or flipped classroom plus exercises (online).

**Examination Information** Written exam

**Prerequisites**

- Calculus
- Linear Algebra

**References**

- U. M. Ascher and C. Greif. *A First Course in Numerical Methods*.
- P. Deufhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing*.
- J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*.
- Lecture notes

Lecture  
6 ECTS

## Operating Systems

**Instructor** Fernando Pedone

**Course Objectives** The goal of this course is to expose students to important topics related to operating systems. Operating systems are a fundamental part of any computer system and prevalent in virtually every application. This course will survey conceptual design issues of such complex programs, and explain the most basic knowledge of operating systems, teaching the difference between the kernel and user modes, and developing key approaches to operating system design and implementation. The course will allow students to understand the implications of some techniques through a hands-on approach.

**Course Description** The course will cover operating systems principles and architecture, process management (i.e., concurrency, process synchronization, threads), memory management (i.e., basic concepts, virtual memory), storage management (i.e., file systems interface and implementation, I/O systems), and operating systems protection and security.

**Learning Methods** Students will learn by following classes, reading suggested material, doing assignments, and working on an operating system project throughout the semester.

**Examination Information** The overall course grade is based on a midterm exam, a course project developed throughout the semester, and a final exam.

**References**

- Operating System Concepts, 10th edition, Silberschatz, Galvin, and Gagne. John Wiley & Sons, 2018.

Atelier  
9 ECTS

## Software Atelier 4: Software Engineering Project

**Instructor** Andrea Mocci,

**Course Objectives** Learn development methodologies, the basics of requirements engineering, testing, and software quality; Learn a minimal set of DevOps skills including CI/CD on GitLab and Docker; Experience project and team development with an iterative, customer-oriented approach.

**Course Description** Programming skills are essential but not enough to develop large and complex software systems that require the coordination of a team of specialists. Software engineering is about the development of such modern software systems. Students will learn to go beyond programming, to coordinate a team, to apply modern methodologies and techniques.

**Learning Methods** The course pursues a learn by doing approach. Students will get a series of theoretical lectures on Software Engineering topics, accompanied by hands on lectures introducing related technologies and frameworks. Students will also develop a project working in teams of ~10 people. Instructors will play the role of the customers, while TAs will play the role of project managers.

**Examination Information** The examination will mainly include the evaluation of a group project for the practical aspects of software engineering, and a final exam that covers the conceptual part of the course.

**Prerequisites**

- Algorithms & Data Structures
- Programming Fundamentals 1
- Programming Fundamentals 2
- Software Atelier 1: Fundamentals of Informatics
- Software Atelier 2: Human-Computer Interaction

**Recommended Courses**

- Programming Fundamentals 3
- Software Atelier 3: The Web

## Bachelor of Science

### Third year

---

Lecture  
3 ECTS

---

## Algorithms & Data Structures 2

---

**Instructor** Evanthia Papadopoulou

---

**Course Objectives** Algorithms and data structures are fundamental to computer science; the performance of any software system depends on the efficiency and correctness of its underlying algorithms and data structures. This course will extend the students' knowledge on fundamental algorithms by considering basic and more advanced topics. It will also strengthen the students' ability to design correct and efficient algorithms on their own.

---

**Course Description** The course covers a variety of topics on algorithms and data structures building upon the material of the first year course "Algorithms and Data Structures". The course will extend the students' knowledge on fundamental algorithms by focusing on several important topics such as data structures for disjoint sets and union-find, interval trees, graph algorithms such as shortest paths, dynamic programming, max-flow/min-cut, introduction to NP completeness.

---

**Learning Methods** Lectures, exercise labs, homework sets on algorithmic problem solving.

---

**Examination Information** The course grade is determined by the results of homework assignments, a midterm exam, and a final exam.

---

**References**

- *Introduction to Algorithms*, 3rd edition, by T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, MIT Press, 2009.

Lecture  
3 ECTS

## Artificial Intelligence

**Instructor** Luca Maria Gambardella

**Course Objectives** Learn and test advanced AI algorithms

**Course Description** Reasoning, learning, searching for new information are key factors in any modern computer system. The goal of this course is to investigate advanced search and meta-heuristics models and algorithms that are useful solve problems of increasing complexity. These are important tools to allow the student to practically and theoretically analyze and solve also real life situations. The course include the AI Cup competition, where students develop and test meta-heuristics algorithms on combinatorial optimization problem instances.

**Learning Methods** Front lecture, text reading, code development

**Examination Information** Mid terms, Final written exam, AI CUP

**References**

- *Artificial Intelligence: a modern approach*. S. Russel and Peter Norvig, Prentice Hall
- Course Material in English will be provided to the students  
Additional readings
- *Artificial Intelligence*, third edition, P.H. Winston, Addison-Wesley
- *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Goldberg, Addison-Wesley, MA

Lecture  
6 ECTS

## Computer Graphics

**Instructor** Piotr Didyk

**Course Objectives** This course is an introduction to the theoretical and practical aspects of computer graphics. It teaches raytracing and rasterization techniques used for synthesizing images, as well as basic concepts of animation for rendering dynamic scenes.

**Course Description** The course starts with an introduction to raytracing, a method for generating pictures of virtual scenes, used for special effects and animated movies. A basic version of such a raytracer is developed in the first week. While learning about the theory of lighting models, texture mapping, etc., the raytracer eventually handles moving objects, shadows, reflections, and refractions. The second half of the course teaches rasterization, an alternative approach used for real-time applications. After learning concepts, students implement their own rasterizer including rendering techniques such as normal mapping and shadow maps.

**Learning Methods** The course is a series of lectures interleaved with interactive classes. The assignments consist of both theoretical and practical assignments.

**Examination Information** The final grade is a result of the grades from the assignments and the final exam.

**References** The course will provide all the materials required for the course. The following are useful, but not required:

- *Fundamentals of Computer Graphics*; Shirley; AK Peters, 2002
- *3D Computer Graphics*; Watt; Addison Wesley, 2000
- *Computer Graphics with OpenGL*; Hearn, Baker; Pearson, 2003
- *Real-time rendering*; Akenine-Moller, Thomas, et al.; 4th AK Peters, 2018
- *OpenGL Reference Manual and Programming Guide*

Lecture  
6 ECTS

## Information Retrieval

**Instructor** Fabio Crestani

**Course Objectives** Nowadays more and more information is available in unstructured or poorly structured form. Examples of information of this type are textual documents, web pages, videos, photos, music, blogs, etc. The goal of this course is to enable the student to understand the foundations of managing unstructured or poorly structured information.

**Course Description** The course aims to assist students to understand techniques for the indexing, retrieval, filtering, clustering, and presentation of textual and multimedia information held in digital archives, the web, and/or multimedia systems. From this perspective the course complements what the student learned from the previous course on Data Management, where only structured information is dealt with.

**Learning Methods** The course consists of theoretical lectures and practical sessions. The practical sessions deal with the design, implementation, and evaluation of an information retrieval system for a small and medium size collection of documents.

**Examination Information** Examination will consist of 3 theoretical tests and 1 individual course project carried out during the semester (no final exam).

**References**

- Required:  
C. Zhai, and S. Massung. *Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining*. ACM Books, 2016.
- Suggested:  
W.B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*, Pearson, 2009.

Lecture  
6 ECTS

## Numerical Computing

**Instructor** Olaf Schenk

**Course Objectives** Numerical computing is an interconnected combination of computer science and mathematics in which we develop and analyze algorithms for solving important problems in science, engineering, medicine, and business -- for example, simulating an earthquake, choosing a stock portfolio, or detecting cancer tumors in medical images. The students will learn principles and practices of basic numerical computation based on seven to eight mini-projects. This is a key aspect of scientific computation.

**Course Description** This class will cover several topics, including: graph clustering, graph partitioning, solving linear systems of equations, page rank algorithm and large-scale nonlinear optimization. As much as possible, numerical methods will be presented in the context of real-world applications.

**Learning Methods** A goal of the course is that students will learn principles and practices of basic numerical methods to enable scientific numerical simulations. This goal will be achieved within six to eight mini-projects with a focus on numerical computing.

**Examination Information** 60% of the grade is determined by mandatory graded mini-projects and 40% is determined by the final written exam during the official examination period.

**References**

- *A First Course in Numerical Methods* by Uri Ascher and Chen Greif, published by the Society for Industrial and Applied Mathematics, available directly from SIAM.
- *Numerical Computing with MATLAB*, C. Moler (available online at <http://www.mathworks.com/moler/chapters.html>) Other material will be passed out as notes.

Atelier  
9 ECTS

## Software Atelier 5: Field Project

<b>Instructor</b>	Michele Lanza, Mauro Prevostini
<b>Course Objectives</b>	The goal of the Field Project Atelier is for the students to obtain hands-on experience with real world problems.
<b>Course Description</b>	Since 2014 the Faculty of Informatics collaborated in this context with 35 companies offering field projects to 162 students. This year we have 10 companies on board. The Field Project Atelier consists of an experience in collaboration with a company and can be done individually or as a group, depending on the given context.
<b>Learning Methods</b>	The Field Project Atelier will take place from 24 September to 17 December 2021, at the company during Thursdays and Fridays the whole day. Important dates to be scheduled are as follows: <ul style="list-style-type: none"> <li>• 23 September: Kick-off meeting at USI</li> <li>• 24 September: Beginning of the field project at the company</li> <li>• 27 October: Update meeting students-professor at USI</li> <li>• 1 December: Update meeting students-professor at USI</li> <li>• 17 December: Closing session at USI</li> </ul>
<b>Examination Information</b>	Pass/Fail Criteria based on the experience and the company feedback
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Algorithms &amp; Data Structures</li> <li>• Calculus</li> <li>• Computer Architecture</li> <li>• Discrete Structures</li> <li>• Linear Algebra</li> <li>• Programming Fundamentals 1</li> <li>• Programming Fundamentals 2</li> <li>• Reason and Responsibility in Decision Making</li> <li>• Software Atelier 1: Fundamentals of Informatics</li> <li>• Software Atelier 2: Human-Computer Interaction</li> <li>• Software Atelier 3: The Web</li> </ul>

Lecture  
3 ECTS

## Experimentation & Evaluation

<b>Instructor</b>	Gabriele Bavota
<b>Course Objectives</b>	To teach the basic skills needed to design and conduct experiments in order to evaluate computer systems or system features.
<b>Course Description</b>	Computer scientists build complex systems or choose among existing systems to satisfy perceived needs and requirements. The system is then deployed in an environment consisting of humans and other systems. How do we know the impact of the system on the environment and how well it meets the perceived requirements? A fundamental skill in informatics is the ability to design experiments for evaluating computer systems from several perspectives. In this course, the students will acquire a basic understanding of how to design such experiments and what pitfalls to avoid during design and experimentation. Basic concepts of experimental design, data measurement, qualitative and quantitative evaluation, and evaluation with and without users will be covered.
<b>Learning Methods</b>	Lectures, presentations, homework assignments and in-class discussions.
<b>Examination Information</b>	The grade is defined based on assignments and final exam.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Probability &amp; Statistics</li> <li>• Programming Fundamentals 2</li> <li>• Software Atelier 2: Human-Computer Interaction</li> <li>• Software Atelier 3: The Web</li> </ul>
<b>References</b>	Will be provided in class.

Lecture  
6 ECTS

## Machine Learning

**Instructor** Cesare Alippi

**Course Objectives** This course is an introduction to computational geometry and its applications. Computational geometry is well related to many application domains, such as pattern recognition, image processing, computer graphics, robotics, geographic information systems (GIS), computer-aided design (CAD), information retrieval, computational science, and others. The students will learn fundamental algorithmic techniques and practice in designing algorithms of their own.

**Course Description** The course covers techniques needed in designing and analyzing efficient algorithms for computational problems in discrete geometry such as convex hulls, triangulations, geometric intersections, Voronoi diagrams, Delaunay triangulations, arrangements of lines and hyperplanes, and range searching. The course covers general algorithmic techniques, such as plane sweep, divide and conquer, incremental construction, randomisation, and approximation, through their application to fundamental geometric problems.

**Learning Methods** Lectures, exercise labs, homework sets on algorithmic problem solving

**Examination Information** Assignments and exams.

**References**

- T.Hastie, R.Tibshirani, J.Friedman, *The elements of statistical learning*, Springer
- Slides and material provided by the professor.

Lecture  
6 ECTS

## Optimization Methods

**Instructor** Rolf Krause

**Course Objectives** Obtain knowledge on: central concepts and ideas of optimization; fundamental optimization techniques, gradient based and gradient free; convex and non-convex optimization; constrained and unconstrained optimization; optimality conditions; introduction to optimization in machine learning.

**Course Description** Optimization is of fundamental importance in virtually all branches of science and technology. Optimization methods find their applications in numerous fields, starting from, e.g., machine network flow and ranging over shape optimization in engineering to optimal control problems and machine learning. This course provides an introduction into the most important methods and techniques in discrete and continuous optimization. We will present, analyze, implement, and test -using illustrative examples- methods for discrete and continuous optimization. Particular emphasis will be put on the methodology and the underlying mathematical as well as algorithmic structure. Starting from basic methods as the Simplex method, we will consider various methods in convex as well as non-convex optimization. This will include optimality conditions, the handling of linear and non-linear constraints, and methods such as interior point methods for convex optimization, Newton's method, Trust-Region methods, and optimal control methods.

**Learning Methods** Lecture, reading, self study, hands-on implementation, discussion, tutorial, written weekly assignments

**Examination Information** There will be a midterm, either as larger project-like assignment or as an written exam. The final exam will be written. The written weekly assignments will also count for the final grade.

- Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, <http://www.deeplearningbook.org>, 2016
- Nocedal Wright, Numerical Optimisation;
- Trust-Region Methods; Conn Gould Toint.
- Practical Methods of Optimisation; R. Fletcher.
- Numerical Optimisation, Series: Springer Series in Operations
- Research and Financial Engineering, Nocedal, Jorge, Wright, Stephen 2nd ed., 2006, XXII, 664 p. 85 illus., [www.springer.com/mathematics/book/978-0-387-30303-1](http://www.springer.com/mathematics/book/978-0-387-30303-1)

Lecture  
6 ECTS

## Theory of Computation

<b>Instructor</b>	Natasha Sharygina
<b>Course Objectives</b>	The class introduces the fundamental mathematical properties of computer hardware, software, and certain applications thereof.
<b>Course Description</b>	The course explores what can and cannot be solved on a computer, how quickly, with how much memory, and on which type of computational model. The class is divided into two major parts: computability theory and complexity theory. Computability theory deals primarily with the question of whether a problem is solvable at all on a computer. Complexity theory considers how efficiently the problem can be solved. Two major aspects are considered: time complexity and space complexity, which respectively address a problem of how many steps does it take to perform a computation, and how much memory is required to perform that computation.
<b>Learning Methods</b>	The theory subjects have strong connections with engineering practice. In addition to theory lectures, the course will offer practical exercises and labs which will involve experimentation with various tools.
<b>Examination Information</b>	There will be a written final exam covering both theory and practice of computability and complexity theories.
<b>Prerequisites</b>	<ul style="list-style-type: none"> <li>• Algorithms &amp; Data Structures</li> <li>• Automata &amp; Formal Languages</li> </ul>
<b>References</b>	<ul style="list-style-type: none"> <li>• <i>Introduction to the Theory of Computation</i>; Michael Sipser, 2006, second edition (Required)</li> </ul>

Lecture  
12 ECTS

## Bachelor Project

**Instructor** Mauro Pezzè

**Course Objectives** The bachelor project gives the student the opportunity to work independently to develop a solution to a significant (i.e., large) problem. The student learns and demonstrates both independence and a systematic approach to problem solving.

**Course Description** The bachelor project gives 12 ECTS, which correspond to the work of 2 typical bachelor-level courses. The students are expected to work throughout the semester under the supervision of their project advisor on the substance of the work, and meet regularly as a group with the Bachelor Project Coordinator to receive instructions about the purpose and mechanics of implementing a long-term project. At the end of the semester, the students produce:

- A project report
- A poster and poster presentation
- A product (if applicable) such as an algorithm, a software library, or application.

**Learning Methods** The learning method is primarily by direct experience. The student works with the advisor's and the co-advisors' supervision. The course helps students find an appropriate topic, carry on the research according to the project schedule, and finalize the report and the poster presentation, by providing guidelines and assistance.

**Examination Information** The grading is an evaluation of the bachelor project committee based on the report and the poster presentation.

**Prerequisites** indicated by the single advisors

## Masters in informatics

4 Semesters' programme  
120 ECTS

---

## Informatics

---

**Directors** Kai Hormann, Evanthia Papadopoulou

---

**Goals and  
contents**

The Faculty of Informatics at Università della Svizzera italiana stands out as a centre of competence in informatics. In a matter of very few years, it has become one of Switzerland's major poles for teaching and research, ranking third after the two Federal Institutes of Technology in Zurich and Lausanne. The Faculty aims to train informatics experts that are interdisciplinary in approach, with abstract thinking and generalization skills, a sound knowledge of Information technologies and their pervasive application domains, as well as project-management and teamwork abilities. The Faculty also offers the opportunity to obtain a double Master's degree in collaboration with the University of Milano-Bicocca.

The Master prepares students for current and emerging technologies in computer science by deepening their theoretical knowledge and sharpening their practical skills. The program is designed for both Bachelor students who wish to complete their education and professionals seeking to refresh their knowledge and sharpen their skills. The Master combines the study of fundamental aspects of computer science with a practical hands-on approach, preparing professionals for successfully pursuing a career in research and development across any application domain.

The Master of Science in Informatics is characterized by a broad offering of topics and subjects that can be freely combined in a learning path tailored to the needs and interests of each student. At USI, students learn how to solve complex problems using modern computing technology. They master the ability to develop automated solutions, introduce them in different business and application domains, and predict and assess their positive impact in the real world. Students can benefit from the research excellence of our Faculty of Informatics by getting involved in ongoing research activities as part of their master thesis project, which can be carried out across the entire second year of the Master.

## Career opportunities

Informatics is both the infrastructure and the engine of today's society. It plays a key role in industry as well as the service sector in Switzerland. The national training and research institutions have acquired a considerable reputation worldwide, in particular in the IT field. Many IT companies have or are planning to have research and development centres in Switzerland. Considering this, graduates in Informatics have excellent opportunities on the job market. The demand for well-educated specialists in Informatics is very high and is expected to grow even more. Graduates of the Master of Science in Informatics are trained to solve complex problems in interdisciplinary areas such as machine learning, intelligent search engines, computer graphics and special effects, computer vision and face recognition, robotics, data science, and they are prepared to design, build, integrate, validate, and maintain reliable and secure software systems for the highly competitive software industry of the 21st century.

## Study plan

Students obtain a Master of Science degree in Informatics by completing the following across the two years:

- 30 out of 36 ECTS of core courses
- 30 ECTS thesis
- 60 ECTS of elective courses, 6 of which can be from non-INF USI Master programmes.

Additionally, they have the possibility to obtain a specialisation in one of the following research areas by completing 18 ECTS of electives and their master thesis in the same area:

- Artificial Intelligence
- Computer Systems
- Geometric and Visual Computing
- Information Systems
- Programming Languages
- Software Development
- Theory and Algorithms

	Course	Instructor	ETCS	
<b>Autumn semester</b>				
Foundational Courses	Algorithms & Complexity	Papadopoulou	6	
	Distributed Systems	Eugster	6	
	High-Performance Computing	Schenk	6	
	Machine Learning	Wand	6	
	Software Engineering	Denaro	6	
Electives*	Advanced Java Programming	Binder, Rosà	6	
	Advanced Topics in Machine Learning	Alippi	3	
	Artificial Intelligence	Gambardella	6	
	Computer Aided Verification	Sharygina	6	
	Deep Learning Lab	Irie	3	
	Distributed Algorithms	Pedone	6	
	Mobile and Wearable Computing	Santini	6	
	Numerical Algorithms	Hormann	6	
	Programming Styles	Hauswirth	3	
	Software Design & Modeling	Furia	6	
	Software performance	Hauswirth	6	
	User Experience Design	Landoni, Langheinrich	6	
	<b>Spring semester</b>			
	Foundational Courses	Information Security	Langheinrich	6
Electives*	Advanced Computer Architectures	Pozzi	6	
	Advanced Networking	Carzaniga	6	
	Business Process Modeling, Management and Mining	Pautasso	3	
	Computational Fabrication	Didyk	6	
	Computer Vision & Pattern Recognition	Hormann	6	
	Geometric Algorithms	Papadopoulou	6	
	Graph Deep Learning	Alippi	3	
	Information & Physics	Wolf	3	
	Quantum Computing	Wolf	6	
	Robotics	Giusti	6	
	Software Analysis	Furia	6	
	Software Architecture	Pautasso	6	
	Software Quality & Testing	Pezzè	6	
	<b>Master thesis **</b>	Faculty	30	
	<b>ETCS Total</b>		120	

\* Including up to 6 ECTS of electives from USI Master courses

\*\* Master Thesis can be started in the 3rd semester.

	Course	Instructor	ETCS
<b>Specialisation in Artificial Intelligence</b>			
Autumn	Advanced Topics in Machine Learning	Alippi	3
	Artificial Intelligence	Gambardella	6
	Deep Learning Lab	Irie	6
Spring	Computer Vision & Pattern Recognition	Hormann	6
	Graph Deep Learning	Alippi	3
	Robotics	Giusti	6
<b>Specialisation in Computer Systems</b>			
Autumn	Computer Aided Verification	Sharygina	6
	Distributed Algorithms	Pedone	6
	Mobile and Wearable Computing	Santini	6
Spring	Advanced Computer Architectures	Pozzi	6
	Advanced Networking	Carzaniga	6
<b>Specialisation in Geometric and Visual Computing</b>			
Spring	Computational Fabrication	Didyk	6
	Computer Vision & Pattern Recognition	Hormann	6
	Geometric Algorithms	Papadopoulou	6
	Graph Deep Learning	Alippi	3
<b>Specialisation in Information Systems</b>			
Autumn	Distributed Algorithms	Pedone	6
	Mobile and Wearable Computing	Santini	6
	User Experience Design	Landoni, Langheinrich	6
Spring	Advanced Networking	Carzaniga	6
	Business Process Modeling, Management and Mining	Pautasso	3
<b>Specialisation in Programming Languages</b>			
Autumn	Advanced Java Programming	Binder, Rosà	6
	Computer Aided Verification	Sharygina	6
	Programming Styles	Hauswirth	3
	Software Performance	Hauswirth	3
Spring	Advanced Computer Architectures	Pozzi	6
	Software Analysis	Furia	6
<b>Specialisation in Theory and Algorithms</b>			
Autumn	Computer Aided Verification	Sharygina	6
	Numerical Algorithms	Hormann	6
Spring	Geometric Algorithms	papadopoulou	6
	Information & Physics	Wolf	3
	Quantum Computing	Wolf	6

## Master of Science in Artificial Intelligence

4 Semesters' programme  
120 ECTS

## Artificial Intelligence

### Directors

Luca Maria Gambardella, Jürgen Schmidhuber

### Goals and contents

Artificial Intelligence (AI) is one of the most popular areas in computer science and engineering. In this master programme a wide variety of techniques will be taught, including intelligent robotics, artificial deep neural networks, machine learning, meta-heuristics optimization techniques, data mining, data analytics, simulation and distributed algorithms. The main courses are integrated with laboratory works where students have the possibility to use real robots and to practice with state of the art tools and methodologies. Artificial Intelligence may not only be the most exciting field in computer science, but of science in general. In fact, the best scientists of the future might even be AIs themselves. Hardware soon will have more raw computational power (CP) than human brains, since CP per cent is still growing by a factor of 100-1000 per decade. And there is no reason to believe that general problem solving software similar to that of humans will be lacking: there already exist mathematically optimal (though not yet practical) universal problem solvers developed at IDSIA. And existing highly practical (but not quite as universal) AI already learn from experience, outperforming humans in more and more fields. For example, biologically plausible deep/recurrent artificial neural networks are learning to solve pattern recognition tasks that seemed infeasible only 10 years ago. Even creativity has been formalized such that it can now be implemented on machines. The current developments in IS may soon lead to the end of history as we know it (more), and as an IS master student you can become part of this revolution. Artificial Intelligence systems have knowledge, beliefs, preferences and goals, and they have informational as well as motivational attitudes. They observe, learn, communicate, plan, anticipate and commit. They are able to reason about other systems and their own internal states, to simulate and optimize their performance. AI systems react to dynamic situations adapting their capabilities through learning mechanisms, with a high degree of autonomy.

## Career opportunities

Students graduating from this programme will develop a taste for working on complex problems. In their future careers they will be able to apply their knowledge in many interdisciplinary areas including robotics, business forecasting, intelligent search, video games, music and entertainment, chat bots, medical diagnostics, self-driving cars, to name a few.

## Study plan

In this master programme a wide variety of techniques will be taught, including intelligent robotics, artificial deep neural networks, machine learning, meta-heuristics optimization techniques, data mining, data analytics, simulation and distributed algorithms. The main courses are integrated with laboratory works where students have the possibility to use real robots and to practice with state of the art tools and methodologies. After the first few lectures of the basic Machine Learning course, AI master students will already know how to train self-learning artificial neural networks to recognize the images and handwritings to the right better than any other known method.

	Course	Instructor	ETCS	
<b>First semester</b>				
Core Courses 18 ECTS	Machine Learning	Wand	6	
	Deep Learning Lab	Irie	3	
	Algorithms & Complexity	Papadopoulou	6	
	Numerical Algorithms	Hormann	3	
Electives 12 ECTS	Advanced Topics in Machine Learning	Alippi	3	
	High-Performance Computing	Schenk	6	
	Introduction to Ordinary Differential Equations	Krause, Pezzuto	3	
	Introduction to Partial Differential Equations	Multerer	6	
	Mobile and Wearable Computing	Santini	6	
	Programming Styles	Hauswirth	3	
	User Experience Design	Landoni, Langheinrich	6	
	Writing Business Plans	Colombo	3	
<b>Second semester</b>				
Core Courses 12 ECTS	Stochastic Methods	Horenko	6	
	Robotics	Giusti	6	
Electives 18 ECTS	Advanced Computer Architectures	Pozzi	6	
	Advanced Networking	Carzaniga	6	
	Business Intelligence and Applications	Martinenghi	6	
	Effective High-Performance Computing & Data Analytics Summer School	Schenk	6	
	Entrepreneurship: Theory & Practice	Colombo	3	
	Geometric Algorithms	Papadopoulou	6	
	Information & Physics	Wolf	3	
	Philosophy and Artificial Intelligence	Facchini, Smith	3	
	Quantum Computing	Wolf	6	
	Solution and Optimization methods for Large Scale Problems	Krause	6	
	<b>Third semester</b>			
	Core Courses 21 ECTS	Artificial Intelligence	Gambardella	6
Distributed Algorithms		Pedone	6	
Master Thesis		Faculty	9	
Electives 9 ECTS	Choose from the electives of the 1st semester			
<b>Fourth Semester</b>				
Core Courses 30 ECTS	Data Analytics	Crestani	6	
	Graph Deep Learning	Alippi	3	
	Master Thesis	Faculty	21	
<b>ETCS Total</b>			120	

## Master of Science in Computational Science

4 Semesters' programme  
120 ECTS

### Computational Science

**Director** Olaf Schenk

#### Goals and contents

Computational Science is the quantitative study of complex systems. The Master of Science degree in Computational Science (MCS) at the Università della Svizzera italiana (USI) offers students the opportunity to acquire an in-depth understanding and set of skills in computational science. Computational science consists of a combination of

- Statistical Data Science
- Applied Mathematics
- High-performance Computing

It offers this innovative combination of methodological competences in data science, algorithmics and computing both to students who have a desire to work as a quantitative analyst in industry and as a computational scientist in academia.

The Master programme has a unique combination of courses from applied mathematics, statistical data science and computer science.

There are additional optional courses from various applications domains, such as finance, biology and sociology, aiming at building deep application-oriented competences in computational science.

Besides methodological skills and applications, the programme focuses on the use and development of scientific simulation and analysis software.

Students will acquire strong competences in abstract thinking within a methodology and application oriented education, which will provide the ability to deal with complex models in various applications areas. Through a wide choice of elective courses enables, students can focus on one of the three methodological foundations, i.e., data science, applied mathematics and high-performance computing, or on a particular field of application. Along with the mentor, each student will set up an individual study plan for selecting the appropriate elective courses. The mentor will advise and accompany the student throughout her/his study.

The multidisciplinary programme offers a streamlined blend of cutting-edge scientific research and practical application, thus

providing an excellent foundation for a corporate, industrial, or academic career. Our students receive a firm grounding in programming, mathematical modeling and numerical simulation. The Master in Computational Science opens the doors to industry in data and software engineering, environmental engineering, financial services, chemical and pharmaceutical R&D. It is also a strong asset for a PhD in computational science.

With the guidance of the Master Director, students will be encouraged to set up an individual study plan that includes appropriate elective courses. The Master Director will advise and accompany students through the entire two-year course of study.

#### Career opportunities

The Computational Science programme offers a streamlined blend of cutting-edge scientific research and practical applications, thus providing an excellent foundation for a corporate, industrial, or academic career. Our students receive a firm grounding in programming, data science, mathematical modeling and numerical simulation. The Master in Computational Science opens the doors to industry in data science, computational engineering, financial services, chemical and pharmaceutical R&D. It is also a strong asset for a PhD in a computational science field, such as applied mathematics, statistics, AI and high-performance computing.

#### Study plan

With the guidance of the Master Director, students will be encouraged to set up an individual study plan that includes appropriate elective courses. The Master Director will advise and accompany students through the entire two-year course of study.

	Course	ETCS	ETCS
<b>First semester</b>			
Core Courses 21 ECTS	Efficient Computational Algorithms	Horenko	6
	High-Performance Computing	Schenk	6
	Introduction to Computational Science	Limongelli, Pivkin, Wit	3
	Introduction to Data Science	Wit	6
Electives 9 ECTS	Advanced Topics in Machine Learning	Alippi	3
	Analysis of Social Networks	Lomi, Stivala	6
	Artificial Intelligence	Gambardella	6
	Bioinformatics	Limongelli	6
	Computational Biology & Drug Design	Limongelli	6
	Deep Learning Lab	Irie	3
	Distributed Algorithms	Pedone	6
	Distributed Systems	Eugster	6
	Introduction to Ordinary Differential Equations	Krause, Pezzuto	3
	Machine Learning	Wand	6
	Numerical Algorithms	Hormann	6
	Software Atelier: Partial Differential Equations	Krause	3
	Software Tools for Computational Science	Limongelli	3

	Course	ETCS	ETCS
<b>Second semester</b>			
Electives 30 ECTS	Advanced Computer Architectures	Pozzi	6
	Advanced Discretization Methods	Pivkin	6
	Advanced Networking	Carzaniga	6
	Computational Fabrication	Didyk	6
	Computer Vision & Pattern Recognition	Hormann	6
	Graph Deep Learning	Alippi	3
	Effective High-Performance Computing & Data Analytics Summer School	Schenk	6
	Functional and Numerical Analysis (FOMICS block course)	Krause	3
	Geometric Algorithms	Papadopoulou	6
	Graph Deep Learning	Alippi	3
	Graphical Models and Network Science	Wit	6
	Information & Physics	Wolf	3
	Introduction to Bayesian Computing	Mira	3
	Introduction to Partial Differential Equations	Multerer	6
	Particle Methods	Pivkin	6
	Quantum Computing	Wolf	6
	Solution and Optimization methods for Large Scale Problems	Krause	6
	Stochastic Methods	Horenko	6
	Text Analysis and Spatial Data for Economists	Parchet, Gessler	6
<b>Third semester</b>			
Mandatory 6 ECTS	Preparation Master's Thesis	Faculty	6
Electives 24 ECTS	Choose from the electives of the 1st semester		24
<b>Fourth Semester</b>			
Mandatory 24 ECTS	Master Thesis	Faculty	24
Electives 6 ECTS	Choose from the electives of the 2nd semester		6
<b>ETCS Total</b>			<b>120</b>

## Master of Science in Informatics and Economics

4 Semesters' programme  
120 ECTS

## Financial Technology & Computing

### Directors

Marc Langheinrich, Erik Nowak, Fernando Pedone, Paul Schneider

### Goals and contents

The Master of Science in Financial Technology and Computing is offered jointly by the Faculty of Informatics and by the Faculty of Economics of the Università della Svizzera italiana. This unique cross-discipline programme combines USI's world-leading expertise in Finance and Informatics and offers exciting career prospects that range from fintech startups to banks and insurers to hedge funds. The programme has been designed to provide graduates with an informatics background with the necessary tools and skills for understanding fundamental problems in finance while, at the same time, learning about advanced tools and techniques in informatics to be applied in finance. The Master of Science in Financial Technology and Computing has been designed to provide graduates with a strong background in informatics with the necessary tools and skills for understanding core challenges in finance while, at the same time, learn about the advanced technology that is needed to drive the next generation finance services. This Master offers a highly challenging programme that delivers key skills in a novel interdisciplinary domain. A two-tiered structure starts students off with a set of well-balanced core courses from both informatics and finance in the first year, followed by a broad set of electives that can be chosen in the second year, according to personal interests and abilities. A fourth semester Master's thesis can then be done either within the context of a university research group, or in collaboration with industry embedded in our Fintech Laboratory. Since English is the sole teaching language, graduates are well-prepared to work in international companies in Switzerland and beyond.

### Career opportunities

The primary labor market for graduates of the programme is to be found in small Fintech startups, medium to large companies in the finance sector (e.g., banks, insurers, hedge funds), as well as the public sector, both in Switzerland and abroad. Many existing financial companies struggle to keep up with recent developments in finance

technology and thus are in great need of informatics professionals who have a thorough understanding of finance. Potential job profiles range from system architect to system developer to service designer to financial consultant. With an MFT master from USI, students will be able to help banks, trading companies, and insurers master this new reality, or, alternatively, be well positioned to challenge existing players with their own startup..

### Study plan

This full time programme stretches over two years. It allows students to personalize their study curricula according to their interests. The core skills are acquired in the first two semesters. The third semester is dedicated to more specialized courses and electives that can be chosen according to the students' preference.

	Course	Instructor	ETCS
<b>First semester</b>			
Core Courses 30 ECTS	Data Analytics for Finance I & II	Gruber	6
	Financial Econometrics**	Mancini	6
	Investments**	Franzoni	6
	Distributed Systems	Eugster	6
	High-Performance Computing	Schenk	6
<b>Second semester</b>			
Core Courses 24 ECTS	Blockchains & Digital Currencies	Morini	3
	Information Security	Langheinrich	6
	Risk Management**	Plazzi	6
	Software Quality & Testing	Pezzè	6
	Trading and Market Microstructure	Kaul	3
Electives 6 ECTS	Financial Intermediation **	Plazzi	6
	Derivatives** (required for "Advanced Derivatives")	Barone Adesi	6

	Course	Instructor	ETCS
<b>Third semester</b>			
Core Courses 18 ECTS	Distributed Algorithms	Pedone	6
	Financial Modelling**	Franzoni	6
	Financial Stability**	Mele	6
	Machine Learning	Wand	6
Electives 12 ECTS	Students choose from electives from the Informatics and Finance courses listed, and from other courses from the Master programmes offered by the Faculty of Informatics and the Faculty of Economics (upon approval of the Master's director).		
Informatics	Algorithms & Complexity	Papadopoulou	6
	Mobile and Wearable Computing	Santini	6
	Software Engineering	Denaro	6
	Software Performance	Hauswirth	6
	User Experience Design	Landoni, Langheinrich	6
Finance	Advanced Derivatives**	Barone Adesi	3
	Alternative Investments	Mueller	6
	Fixed Income Markets**	Mele	6
	Launching Fintech Ventures	Cavatore	3
	Project Management	Gonçalves	3
	Qualitative Marketing Research and Data Analysis	Pellandini-Simányi	3
<b>Fourth Semester</b>			
Core Course 30 ECTS	Master Thesis *	Faculty	30
Electives* 6/0 ECTS	Students choose from electives from the Informatics and Finance courses listed, and from other courses from the Master programmes offered by the Faculty of Informatics and the Faculty of Economics (upon approval of the Master's director).		
Informatics	Graphical Models and Network Science	Wit	6
	Software Architecture	Pautasso	6
	Software Atelier: Simulation, Data Science & Supercomputing	Schenk	6
	Critical Consumer Behaviour	Gibbert	6
<b>ETCS Total</b>			120

\* The Master thesis can be started in the 3rd semester (6 ECTS)

\*\* To obtain the SFI accreditation, students have to achieve 45 ECTS among these courses.

## Master of Science in Informatics and Economics

4 Semesters' programme  
120 ECTS

### Management & Informatics

#### Directors

Marc Langheinrich, Nikolaus Beck

#### Goals and contents

The Master of Science in Management & Informatics has been designed to provide graduates from a wide variety of backgrounds (informatics, economics, mathematics, business, engineering, etc.) with the necessary tools and skills for understanding complex information technology (IT) problems while, at the same time, knowing about the needs and requirements of a modern organization. This Master offers a balanced combination of courses that cover the necessary background in management, fundamental aspects of current and evolving IT, as well as specialized topics at the interface between management and informatics, such as enterprise resource planning. Since the programme is taught entirely in English, graduates are well prepared to work in international companies. Moreover, the interdisciplinary approach of this Master provides a general skill to work across traditional areas.

This full-time programme stretches over two years. It allows students to personalize their study curricula according to their interests. The first year focuses on the acquisition of foundational knowledge. Students who obtained a Bachelor's degree in informatics or a related field (mathematics, engineering, physics, etc.) enter the programme in the "Informatics track" and follow a set of courses that provide them with a solid background in management disciplines. In contrast, the "Management track" targets students with a background in economics or management, and teaches the fundamental principles of informatics. In addition, all students attend mandatory courses that cover the interface between management and informatics.

The second year offers specialized courses and electives that students can choose according to their preferences. A mandatory practical field project lets student gain practical consulting experience by working for real clients in small project teams. A substantial master's thesis concludes the programme.

#### Career opportunities

Graduates from this Master will acquire a broad background in Informatics, allowing them to easily interact, on a technical level, with the IT department of an organization, both to evaluate technical proposals as well as to articulate possible solutions to the organization or the customer. On the other hand, graduates will also understand the tactical and strategic use of IT to enhance the efficiency of an organization, or how to explain user requirements in terms that can be understood by the IT department or the client. Most companies struggle with integrating IT in their organization, so people who can be the interface between the technical and organizational parts of a company are in great demand. Graduates of the programme will find work in medium to large companies, as well as the public sector, both in Switzerland and abroad.

#### Study plan

This full time programme stretches over two years. It allows students to personalize their study curricula according to their interests. The basic knowledge is acquired in the first two semesters. Students who obtained a Bachelor's degree in informatics or a related field (mathematics, engineering, physics, etc.) enter the programme in the Informatics track and follow a set of courses that provide them with a fundamental insight into the management disciplines. In contrast, the Management track is tailored for students with a background in economics or management and teaches the basic aspects of informatics. In addition, all students attend mandatory courses that cover the interface between management and informatics. The third and fourth semester are dedicated to specialized courses and electives that can be chosen according to the students' preference. Moreover, the students participate in a practical field project, which is done in groups for a real company, and conclude their studies by writing a substantial master's thesis.

	Course	Instructor	ETCS
<b>First semester</b>			
Core Courses 12 ETCS	Enterprise Resource Planning	Cappiello	6
	Enterprise Resource Planning Lab	Plebani	3
	Project Management	Gonçalves	3
Informatics track 18 ETCS	Corporate Strategy	Prato	6
	Financial Accounting	Tettamanzi	3
	Managerial Accounting	Seistratkova	3
	Orthodox and Critical Perspectives in Marketing	Mendini, Visconti	6
Management track 18 ETCS	Databases	Crestani	6
	Introduction to Programming	Binder	6
	Probability & Statistics	Wit	6
<b>Second semester</b>			
Core Courses 12 ETCS	Business Dynamics	Gonçalves	3
	Business Intelligence and Application	Martinenghi	6
	Business Process Modeling, Management and Mining	Pautasso	3
	Entrepreneurship: Theory and Practice	Colombo	3
	Operations Management	Dietl	3
Informatics track 6 ETCS	Decision Making	Martignoni	3
	Entrepreneurship: Theory and Practice	Colombo	3
Management track 6 ETCS	Fundamentals of Informatics	Sharygina	6
Electives 12 ETCS	Critical Consumer Behaviour	Gibbert	6
	Human Resources Management	Solari	3
	Innovation	Vijay Munshi	3
	International Business	Ciabuschi	3
	Organizational Learning	Beck	6
	Information Security	Langheinrich	6
	Robotics	Giusti	6
	Software Quality & Testing	Pezzè	6
	Text Analysis and Spatial Data for Economists	Parchet, Gessler	6

	Course	Instructor	ETCS
<b>Third semester</b>			
Core Courses 6 ETCS	Lean Six Sigma	Rossetti	6
	Field Project	Langheinrich	12
Capstone Work 12 ETCS			
Electives 12 ECTS	Analytical Thinking	Beck	3
	Business Analytics	Martignoni	3
	Digital Challenges in Marketing and Big Data	Mandelli	3
	Organizational Design & Change	Pallotti	3
	Service Design Marketing	Mandelli	3
	Distributed Systems	Eugster	6
	Machine Learning	Schmidhuber, Wand	6
	Mobile and Wearable Computing	Santini	6
	User Experience Design	Landoni, Langheinrich	6
	<b>Fourth Semester</b>		
Capstone Work 18 ETCS	Master Thesis	Faculty	18
Electives 12 ECTS	Choose from the electives of the 2nd semester		12
<b>ETCS Total</b>			120

## Master of Science in Software & Data Engineering

4 Semesters' programme  
120 ECTS

### Software & Data Engineering

#### Directors

Cesare Pautasso, Gabriele Bavota

#### Goals and contents

Software plays a pivotal role in almost all aspects of our life, including transportation, communication, economy, and health-care. We put trust in software to accomplish complex and vital tasks for us, such as managing our finances, sharing our family and friends' memories, diagnosing diseases, flying airplanes or driving cars. The complexity of these tasks, while becoming transparent to us, does not go away: it is distilled into the software our civilization depends on. Indeed, we are already in the era of ultra-large-scale software systems, composed by millions of code components interacting among them. In such a scenario, software cannot be understood without its data and data becomes valuable only thanks to the software analyzing it. In other words, software engineering aims at managing the complexity of software, keeping it under control. Data engineering focuses instead on how to collect, store, and process huge amounts of data, that can be analyzed to gather insights and support decision making activities. The master features courses taught by world's leading researchers of the Software Institute at the USI Faculty of Informatics.

#### Career opportunities

Data is the new natural resource to be mined and exploited using software. Data analytics software provides actionable insights at the basis of continuous improvement and decision making processes. Such insights can be found by exploring large quantities of data, by asking the right questions and knowing how to reliably and efficiently find the appropriate answers. Students graduating in this Master are highly specialized software and data engineers, with high employability in industry, who are able to fully understand and manage the complexity of modern software systems and of the ocean of data surrounding them. Mastering how to effectively use software to deal with the data deluge is a key capability for any organization undergoing digital transformation efforts. The demand for software and data engineers is currently very high and it is expected to grow even more in the near

future across multiple industry domains (finance, energy, transportation, health, and media).

#### Study plan

The study programme consists of four modules: Software Engineering, Data Engineering, Electives, and Master thesis. The Software Engineering module embraces 30 ECTS and provides students with a deep knowledge of state-of-the-art techniques. Topics related to this module are software design, software architecture, software performance, software analysis, domain-specific languages, and programming styles. The Data Engineering module includes three courses (18 ECTS) teaching students techniques and tools to design and model data (1st semester), to convert data into information (2nd semester), and to transform information into knowledge useful to support decision making activities (3rd semester). The topics studied in the Software and the Data Engineering modules are continuously integrated through the whole course of study. This is done by devoting 18 ECTS to deal with both Software and Data Engineering with project based learning during the Design 101, the Visual Analytics and the Software Analytics atelier. The Electives module includes 18 ECTS, that the student can freely select from a given list of courses offered at the USI Faculty of Informatics based on his/her personal preference. Finally, the remaining 36 ECTS are dedicated to the MSc thesis. Students will use the 6 ECTS of the 3rd semester to visit the research groups of the Software Institute of the Faculty of Informatics and to prepare a thesis proposal. Then, they will work full time on the thesis in the 4th semester in the research group of their choice.

	Course	Instructor	ETCS
<b>First semester</b>			
Mandatory 24 ECTS	Data Design & Modeling	Brambilla	6
	Engineering of Domain Specific Languages	Mocci	3
	Programming Styles	Hauswirth	3
	S&DE Atelier: Design 101	Lanza, Cleve	6
	Software Design & Modeling	Furia	6
Electives 6 ECTS	Cyber-Physical Software Engineerin	Banzi, Ferrante	6
	Distributed Systems	Pedone	6
	High-Performance Computing	Schenk	6
	Mobile and Wearable Computing	Santini	6
	Software Engineering	Denaro	6
<b>Second semester</b>			
Mandatory 24 ECTS	Information Modeling & Analysis	Tonella	6
	S&DE Atelier: Visual Analytics	D'Ambros	6
	Software Analysis	Furia	6
	Software Architecture	Pautasso	6
Electives 6 ECTS	Advanced Networking	Carzaniga	6
	Information Security	Langheinrich	6
	Software Quality & Testing	Pezzè	6
<b>Third semester</b>			
Mandatory 24 ECTS	Knowledge Analysis & Management	Tonella	6
	S&DE Atelier: Software Analytics	Bavota	6
	Software & Data Engineering Seminar	Faculty SI	6
	Software Performance	Hauswirth	6
<b>Fourth Semester</b>			
Mandatory 30 ECTS	Master Thesis	Faculty	30
<b>ETCS Total</b>			<b>120</b>

## Master specialistico per l'insegnamento dell'informatica

6 Semesters' programme  
107 ECTS

## Master specialistico per l'insegnamento dell'informatica

**Direttori** Antonio Carzaniga, Matthias Hauswirth

**Descrittivo** Programma di studi per la formazione di docenti di liceo qualificati per l'insegnamento dell'informatica come materia di Maturità secondo i nuovi piani di studio che, per l'Ordinanza sulla Maturità del Consiglio Federale, dovranno entrare in vigore nell'anno accademico 2022/23.

Ulteriori informazioni: <https://search.usi.ch/educations/81/master-specialistico-per-linsegnamento-dellinformatica>

---

PhD programme

The PhD programme of the Faculty of Informatics at the Università della Svizzera italiana promotes the development of professionals interested in academic or industrial research. A successful PhD student will gain a broad knowledge and understanding of the general field of informatics, as well as an in-depth specialisation in an area of interest. Working with one or more members of the Faculty, who serve as the student's advisors, the student will learn the methods and practical skills to conduct research, and will contribute original, useful, and scientifically valid ideas in their chosen area of interest. PhD students are also encouraged to explore other areas and to interact and collaborate with other students and professors within the Faculty as well as in the broader research community. At present the Faculty awards the following qualifications: PhD in Informatics and PhD in Computational Science. Most students in the PhD programme are supported as assistants. The support covers tuition and provides a stipend. Responsibilities of assistants include both teaching and research duties. Generally students receive support as long as funds are available and the student is making adequate progress through the programme (as described in the regulations). The PhD programme is governed by regulations adopted by the Faculty: [www.inf.usi.ch/regolamenti\\_tutti.htm](http://www.inf.usi.ch/regolamenti_tutti.htm) In order to be admitted, the applicant must have completed a Masters degree in computer science, informatics, or a closely related field prior to joining the programme (but not necessarily prior to applying to the programme). For more information regarding the admission to the programme: [www.inf.usi.ch/dottorato\\_regolamenti.htm](http://www.inf.usi.ch/dottorato_regolamenti.htm)

**Study plan**

The Faculty of Informatics offers PhD courses to students pursuing a PhD at the Faculty. The course Introduction to Doctoral Studies is mandatory for first year PhD students. Master courses may be cross-listed as PhD courses.

---

## Lecturers' profiles

**Marc Langheinrich**

Prof. Langheinrich received his PhD (Dr. sc. ETH) on the topic of "Privacy in Ubiquitous Computing" from the ETH Zurich, Switzerland, in 2005. He has published extensively on both privacy and usability of ubiquitous and pervasive computing systems, and is a regular program committee member of various conferences and workshops in the areas of pervasive computing, security and privacy, and usability. Marc is an Associate Editor in Chief of IEEE Pervasive Computing Magazine, and a Steering Committee member of the UbiComp and IoT conference series. In 1997, Marc spent a year as a Fulbright Scholar at the University of Washington, Seattle, where he also completed his thesis work in the fields of information retrieval and software agents. From 1997 to 1999 he lived in Tokyo, Japan, where he worked at NEC Research on projects involving personalization and electronic commerce. Prof. Langheinrich joined USI in 2008.  
marc.langheinrich@usi.ch

**Cesare Alippi**

Cesare Alippi was awarded the degree in electronic engineering cum laude and PhD from Politecnico di Milano, Italy. IEEE Fellow; Board of Governors member, International Neural Network Society; Board of Directors member, European Neural Network Society; Past Vice-President, IEEE Computa-

al Intelligence Society; AE IEEE Computational Intelligence Magazine, IEEE-Trans. Instrumentation and Measurements, IEEE-Trans. In 2016 he received the International Neural Networks Society Gabor Award and the Outstanding IEEE Transactions on Neural Networks and Learning Systems Paper Award; the 2013 IBM Faculty award; the 2004 IEEE Instrumentation and Measurement Society Young Engineer Award. Current research activity addresses adaptation and learning in non-stationary environments and Intelligence for embedded, cyber-physical systems and IoT. He holds 5 patents, has published one monograph book, 6 edited books and about 200 papers in international journals and conference proceedings.  
alippi@elet.polimi.it

**Walter Binder**

Walter Binder is a professor in the Faculty of Informatics, Università della Svizzera italiana (USI), Switzerland. He holds an MSc, a PhD, and a Venia Docendi from TU Wien, Austria. Before joining USI, he was a post-doctoral researcher in the Artificial Intelligence Laboratory, École Polytechnique Fédérale de Lausanne, Switzerland. His main interests are in the areas of program analysis, virtual machines, parallel programming, and Cloud computing.  
walter.binder@usi.ch

**Michael Bronstein**

Michael Bronstein received the Ph.D. in computer science (2007) from the Technion in Israel. His main research interests are geometric methods in computer vision, pattern recognition, and computer graphics. Prof. Bronstein's research was featured in international news and recognized by several awards, including three ERC grants, Google faculty award, Radcliffe fellowship from Harvard University and Rudolf Diesel fellowship from TU Munich. He has served on program committees of major conferences in his field and was keynote speaker in numerous international symposia. Prof. Bronstein is also actively involved in technology transfer and consulting. His start-up track record includes Novafora (2004-2009 as co-founder and VP of video technology) and Invision (2009-2012 as one of principle technologists). Since the acquisition of Invision by Intel in 2012, Michael has also served as a Research Scientist and Principal Engineer at Intel, where he was one of the key algorithm developers for the RealSense 3D sensor. michael.bronstein@usi.ch

**Antonio Carzaniga**

Antonio Carzaniga joined the Faculty of Informatics at USI when the Faculty was founded in 2004. From 2001 to 2007 he was also an assistant research professor in the Department of Computer Science at the University of Colorado at Boulder, USA. Antonio received the Laurea degree in electronics engineering and the Ph.D. degree in computer science from Politecnico di Milano, Italy. Antonio is a curious researcher. His primary research interests are in the areas of distributed systems and software engineering, specifically in content-based networking, information-centric networking, distributed publish/subscribe systems, middleware, software adaptability and automatic fault tolerance, and testing. He also conducted research in software configuration management and code mobility. Antonio is also a dedicated and passionate teacher. He has developed and taught a

number of courses in the Faculty of Informatics at USI, including Algorithms and Data Structures, Computer Networking, and Systems Programming. antonio.carzaniga@usi.ch

**Fabio Crestani**

Fabio Crestani is a Full Professor at USI since 2007. Previously he was Professor (2000-06) at the University of Strathclyde (UK) and Assistant Professor (1992-97) at the University of Padua (Italy). In between he was Research Fellow at the University of Glasgow (UK), at the International Computer Science Institute in Berkeley (USA), and at the Rutherford Appleton Laboratory (UK). Recently he received a Chair of Excellence at the University Carlos III in Madrid (2011-12), a Visiting Scholarship at Yahoo! Labs (2014), and a Visiting Professorship at the UPMF in Grenoble (2015). Fabio holds a degree in Statistics and Economics from the University of Padua (Italy) and a MSc and PhD in Computing Science from the University of Glasgow (UK). He leads the local Information Retrieval and Text Mining group (see <http://www.ir.inf.usi.ch/> for details). fabio.crestani@usi.ch

**Patrick Eugster**

Patrick Eugster joined USI as a Full Professor Computer Science in 2017. He is also an Adjunct Faculty at Purdue University (where he was a regular faculty member 2005-2016) and at TU Darmstadt (2014-2017). Patrick holds M.S. (1998) and Ph.D. (2001) degrees from EPFL. Patrick is interested in software systems, with a particular focus on distributed systems and programming models/languages, and the intersection between the two. He has co-authored over 120 scientific articles on these topics. His research has been awarded by various funding agencies (e.g., US NSF CAREER 2007, DARPA Computer Science Study Group 2011, ERC Consolidator 2014) and companies (e.g., Google Research Award 2003, NetApp Faculty Fellowship 2014). patrick.eugster@usi.ch

**Luca Maria Gambardella**

Luca Maria Gambardella is director of IDSIA, Dalle Molle Institute for Artificial Intelligence, a joint research institute between USI and SUPSI in Manno. He is Professor at SUPSI (Algorithms course) and Adjunct Professor at the Informatics Faculty at USI (Artificial Intelligence course). He is leading the Intelligent Systems Master Research Unit at SUPSI. His major research interests are in the area of optimization, swarm robotics and multi-agent learning, applied to academic and real-world problems. In particular, he has invented influential and frequently cited Ant Colony Optimization algorithms for combinatorial optimization problems. He is leading projects for the Swiss NSF, Swiss CTI, EU Commission and Industries. To date (Feb 2016) he has published more than 275 publications. According to Google scholar his h-index is 57 and his articles have received more than 34200 citations. He is president of the Swiss Operations Research Society and it was president of the Euro Excellence In Practice Award 2015. luca.gambardella@usi.ch

**Ilia Horenko**

Ilia Horenko is full professor in the faculty of informatics and the Institute of Computational Science of the Università della Svizzera italiana. He received a Ph.D. in applied mathematics from the Free University (FU) of Berlin in 2004 and spent several years as a postdoctoral research fellow at the Biocomputing Group and Climate Research Group at the FU Berlin, before joining the faculty of mathematics and computer science of the FU Berlin as an assistant professor in 2008. His research interests are focussed on the development and practical implementation of data analysis algorithms and time series analysis approaches. Published applications of the methods developed by I. Horenko include problems from climate research, economics, biophysics, engineering and bioinformatics. Prof. Horenko has published over 50 papers in the professional literature. He was a co-organizer of several big

scientific programs and is a frequent reviewer for international funding agencies and the top journals in his field. illia.horenko@usi.ch

**Kai Hormann**

Kai Hormann is a full professor in the Faculty of Informatics at USI. He received a PhD in computer science from the University of Erlangen in 2002 and spent two years as a postdoctoral research fellow at Caltech in Pasadena and the CNR in Pisa, before joining Clausthal University of Technology as an assistant professor in 2004. During the winter term 2007/2008 he visited Freie Universität Berlin as a BMS substitute professor and came to Lugano as an associate professor in 2009. His research interests are focussed on the mathematical foundations of geometry processing algorithms and their applications in computer graphics and related fields. In particular, he is working on generalized barycentric coordinates, subdivision of curves and surfaces, barycentric rational interpolation, and dynamic geometry processing. Professor Hormann has published over 70 peer-reviewed papers and is an associate editor of Computer Aided Geometric Design, Computers & Graphics, and the Dolomites Research Notes on Approximation. kai.hormann@usi.ch

**Rolf Krause**

Rolf Krause is chair of advanced scientific computing and the director of the institute of computational science in the faculty of informatics. From 2003 to 2009, he was professor at the University of Bonn. During that time he spent a sabbatical at UC San Diego (USA) and Columbia University New York (USA). In 2002 he was on a Postdoctoral research visit at the Courant Institute (NYU, New York). He holds a Diploma and a PhD (2000) in Mathematics from the FU Berlin (Germany). His research focuses on numerical simulation and mathematical modeling in the life sciences, in particular medicine, and for engineering applications. A focal point of his work is the

development and implementation of parallel simulation-methods, which show excellent performance also in real world applications. He is member of the editorial board of the SIAM Journal on Scientific Computing (SISC) and of the Springer Journal Computing and Visualization In Scienc (CVS).  
rolf.krause@usi.ch

#### Michele Lanza

Michele Lanza is full professor at the faculty of informatics of the Università della Svizzera italiana, Switzerland, which he co-founded in 2004. His doctoral dissertation, completed in 2003 at the University of Bern, received the Denert award for best thesis in software engineering of 2003. Prof. Lanza received the Credit Suisse Award for best teaching in 2007 and 2009. At the Università della Svizzera italiana Prof. Lanza directs the Software Institute (<http://si.usi.ch>) and leads the REVEAL research group, working in the areas of software visualization, evolution, and analytics. He authored more than enough articles and the book "Object-Oriented Metrics in Practice". Prof. Lanza is involved in a number of scientific communities, and has served on close to a 100 program committees. He likes (software) visualization, universal design, and when it comes to programming he thinks it's nothing but Smalltalk.  
michele.lanza@usi.ch

#### Evanthia Papadopoulou

Evanthia Papadopoulou is a professor of computer science at the Università della Svizzera italiana. From 1996 to 2008 she had been a research staff member at the IBM T.J. Watson research center, Yorktown Heights NY, USA. She had also been a Faculty member in the department of computer science at the Athens University of Economics and Business. She holds a BS degree in mathematics from University of Athens, an MS in computer science from University of Illinois at Chicago, and a Ph.D. in computer science from Northwestern University, USA, December 1995. Her research interests include the design and analysis of discrete

algorithms, computational geometry and its applications, robust geometric computing and the implementation of geometric algorithms, algorithmic aspects of VLSI computer-aided design. For her work on "Voronoi diagram based VLSI Critical Area Analysis", she received the IBM outstanding innovation award, August 2006, and the Technical Accomplishment for IBM Research, December 2006  
evanthia.papadopoulou@usi.ch

#### Cesare Pautasso

Cesare Pautasso is full professor at the USI Faculty of Informatics. Previously he was a researcher at the IBM Zurich Research Lab and a senior researcher at ETH Zurich, where he earned his PhD in 2004. His research group is interested in interdisciplinary research around all aspects of the architecture, design and engineering of next-generation Web information systems, focusing on building experimental systems to explore the intersection of model-driven software composition techniques, RESTful business process management, liquid software architectures, and edge computing. He edits the IEEE Software Insights department. He was the general chair of the 16th International Conference on Web Engineering (ICWE2016) held in Lugano in June 2016. He regularly referees for Swiss, EU and International funding agencies. Since 2010 he is an advisory board member of Enterprise Web. You can follow him on [@pautasso](http://design.inf.usi.ch) and [@pautasso@scholar.social](mailto:@pautasso@scholar.social)  
cesare.pautasso@usi.ch

#### Fernando Pedone

Fernando Pedone is a full professor in the Faculty of Informatics at the Università della Svizzera italiana (USI), Switzerland, and has been also affiliated with Cornell University (as a visiting professor), EPFL, and Hewlett-Packard Laboratories (HP Labs). He received the Ph.D. degree from EPFL in 1999. His research interests include the theory and practice of distributed systems and distributed data management

systems. Fernando Pedone has authored more than 100 scientific papers and 6 patents. He is co-editor of the book "Replication: theory and practice", Springer 2010.  
fernando.pedone@usi.ch

#### Mauro Pezzè

Mauro Pezzè is a professor of software engineering at the Faculty of Informatics of USI. Mauro Pezzè has served as associate editor of IEEE Transactions on Software Engineering from 2013 to 2018 and has served as associate editor of ACM Transactions on Software Engineering and Methodology from 2006 to 2012. He has been program co-chair of the International Conference on Software Engineering (ICSE 2012), general chair (2013) and program chair (2006) of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), program chair of the International Conference on Fundamental Approaches to Software Engineering (ETAPS-FASE 2003). Pezzè's research interests are in software engineering and in particular in the areas of software testing and analysis and self-healing and autonomic systems. He is the co-author of more than 160 papers appeared in the top journals and in the proceedings of the main conferences in software engineering and of a book on Software Testing and Analysis.  
mauro.pezze@usi.ch

#### Igor Pivkin

Igor Pivkin received his B.Sc. and M.Sc. degrees in Mathematics from Novosibirsk State University, Russia, M.Sc. degree in Computer Science and Ph.D. in Applied Mathematics from Brown University, USA. Before coming to Lugano, he was a Postdoctoral Associate in the Department of Materials Science and Engineering at Massachusetts Institute of Technology, USA. The research interests of Igor Pivkin lie in the area of multiscale/multiphysics modeling, corresponding numerical methods and parallel large-scale simulations of biological and physical systems. Specific areas include biophysics, cellular and

molecular biomechanics, stochastic multiscale modeling, and coarse-grained molecular simulations.  
igor.pivkin@usi.ch

#### Laura Pozzi

Laura Pozzi joined the Faculty of Informatics at USI as Assistant Professor in 2005, and she is now a Full Professor. Before landing at USI, Laura was a postdoctoral researcher in the Processor Architecture Laboratory of EPFL, a research engineer with STMicroelectronics in the Silicon Valley in California, and an Industrial Visitor at UC Berkeley. Laura holds a Master (1996) and a Ph.D. degree (2000) in Computer Engineering from Politecnico di Milano, Italy. Her research interests include embedded processor customization, compiler techniques for customizable processors, coarse-grain reconfigurable fabrics, and in general application-mapping onto unconventional architectures. Laura currently teaches two courses at USI: Automata and Formal Languages in the Bachelor; Advanced Computer Architectures in the Master.  
laura.pozzi@usi.ch

#### Olaf Schenk

Olaf Schenk is a full professor at the Institute of Computational Science within the Faculty of Informatics at the Università della Svizzera italiana, Switzerland. He graduated in Applied Mathematics from Karlsruhe Institute of Technology (KIT), Germany, and earned his PhD from the Department of Information Technology and Electrical Engineering of ETH Zurich. He conducts research in applied algorithms, computational science, and software tools for high-performance scientific computing. He is a recipient of an IBM faculty award (2008) and two leadership computing awards from the U.S. Department of Energy (2012, 2013). He serves on the editorial board of the SIAM Journal for Scientific Computing and has been elected as the Program Director for the SIAM Activity group on Supercomputing (2016-2017) and as the current Vice Chair for the

SIAM Activity group on Supercomputing (2018-2019). He regularly is a chair of international computer science conferences (e.g. ACM PASC, SIAM PP, IEEE CSE).olaf.schenk@usi.ch

#### Jürgen Schmidhuber

Since age 15 or so, Prof. Jürgen Schmidhuber's main scientific ambition has been to build an optimal scientist through self-improving Artificial Intelligence, then retire. His AI team has won nine international competitions in machine learning and pattern recognition (more than any other AI research group worldwide) and six independent best paper/best video awards, achieved the world's first superhuman visual classification results, and established the field of mathematically rigorous universal AI and optimal universal problem solvers. His formal theory of creativity & curiosity & fun explains art, science, music, and humor. He generalized algorithmic information theory and the many-worlds theory of physics. Many famous leading companies are now using the machine learning techniques developed in his group at the Swiss AI Lab IDSIA (a Business Week Top 10 AI Lab) & USI & SUPSI. He published 333 peer-reviewed papers, and is recipient of the 2013 Helmholtz Award of the International Neural Networks Society. juergen@idsia.ch

#### Natasha Sharygina

Prof. Sharygina's research focuses on improving the program development process through formal methods of specification and verification. In particular, her interests lie in automated formal verification with a specific focus on software/hardware model checking, information security, static analysis, abstract interpretation and decision procedures. Prof. Sharygina received a Ph.D. degree from the University of Texas at Austin, USA in 2002. Her professional experience includes consulting at Bell Labs, Lucent Technologies Computing Sciences Research in 2000-2001 and a research faculty position at Carnegie Mellon University, SEI in 2002-2005. Prof.

Sharygina directs the USI Formal Verification and Security group ([www.verify.inf.usi.ch](http://www.verify.inf.usi.ch)) whose work resulted in award-winning theoretical frameworks and practical tools to enable sound and scalable verification of industrial-size systems. natasha.sharygina@usi.ch

#### Paolo Tonella

Paolo Tonella is Full Professor at the Faculty of Informatics and at the Software Institute of Università della Svizzera Italiana (USI) in Lugano, Switzerland. He is also Honorary Professor at University College London, UK. Until mid 2018 he has been Head of Software Engineering at Fondazione Bruno Kessler, Trento, Italy. Paolo Tonella holds an ERC Advanced grant as Principal Investigator of the project PRECRIME. Paolo Tonella wrote over 150 peer reviewed conference papers and over 50 journal papers. His H-index (according to Google scholar) is 52. He is/was in the editorial board of the ACM Transactions on Software Engineering and Methodology, of the IEEE Transactions on Software Engineering, of Empirical Software Engineering, Springer, and of the Journal of Software: Evolution and Process, Wiley. Paolo Tonella teaches Information Modeling and Analysis at the Master in Data and Software Engineering. paolo.tonella@usi.ch

#### Ernst Wit

Ernst Wit joined the Faculty of Informatics at USI as full professor in June 2018. He obtained PhDs in Philosophy (1997) and Statistics (2000) in the US. From 2000 until 2005 he was in the Statistics Department at the University of Glasgow, where he became a Reader. In 2005 he became head of the Medical Statistics Unit (12 FTE) at the University of Lancaster as full professor and as Director implemented a thriving Master in Statistics programme. From 2008 to 2018 Wit was at the University of Groningen, where he continued to work on methodological development in high-dimensional inference with a specific focus on network modelling. He is the author of 105

peer-reviewed publications, and has served as the President of the European Bernoulli Society and as member of the Board of Directors of the International Biometrics Society. He advises the Ministry of Internal Affairs in the Netherlands on statistical matters relating to elections and referendums since 2014. ernst.wit@usi.ch

#### Stefan Wolf

Stefan Wolf joined USI's Informatics Faculty in 2011 as an Associate Professor, and is Full Professor since 2015. Born in Schaffhausen, Switzerland, he received a Dipl. Math. ETH, followed by a Ph.D. in Computer Science in the field of Information-Theoretic Cryptography from ETH Zurich under the supervision of Professor Ueli Maurer. After a postdoc at McGill University, Montreal, he was Assistant Professor at University of Waterloo, Ontario, and Université de Montréal, Quebec. From 2005 to 2011, he was a Swiss National Science Foundation (SNF) Professor for Quantum Information at the Computer Science Department of ETH Zurich. His research domain lies in the fields of cryptography, information theory, and quantum information processing. stefan.wolf@usi.ch

**Gabriele Bavota**

Gabriele Bavota received (cum laude) the Laurea in Computer Science from the University of Salerno (Italy). He received the PhD in Computer Science from the University of Salerno in 2013. From January 2013 to October 2014 he has been a research fellow at the University of Sannio. From November 2014 to August 2016 he was Assistant Professor at Free University of Bolzano-Bozen. He joined the Faculty of Informatics of USI as an Assistant Professor in September 2016. His research interests include software maintenance and evolution and empirical software engineering. On these topics, he authored over 100 papers appeared in international journals and conferences. He is the recipient of the 2018 ACM Sigsoft Early Career Researcher Award for outstanding contributions in the area of software engineering as an early career investigator.  
gabriele.bavota@usi.ch

**Carlo A. Furia**

Carlo A. Furia is an associate professor at the Software Institute in the Faculty of Informatics of the Università della Svizzera Italiana (USI). His research interests center around developing rigorous yet practical techniques and tools to help improve the quality, correctness, and reliability of software and systems; much of his work aims at making verification and formal methods practical and widely applicable. He publishes in software

engineering, formal methods, and verification venues. He has a PhD in computer science from the Politecnico di Milano; Before joining USI, he was an associate professor at Chalmers University of Technology. Before Chalmers, he spent several years as senior researcher at ETH Zurich.  
carlo.alberto.furia@usi.ch

**Matthias Hauswirth**

After an apprenticeship as an "Elektroniker" in Thun, Matthias Hauswirth received an "Informatik Ing. HTL" from Ingenieurschule Biel and later a Masters and a PhD from the University of Colorado at Boulder. During his studies he spent several summers as an intern at Microsoft Research and at the IBM T. J. Watson Research Center. Right after his graduation, in Fall 2005, he joined the Faculty of Informatics here at USI. In 2015/2016 he spent a sabbatical at SRI International to learn more about learning. Matthias is interested in the area between programming languages, runtime systems, and software engineering, with a focus on the efficiency of software, its users, and its developers. He loves to help students learn, and he is interested in finding approaches to increase the efficiency and the effectiveness of teaching informatics.  
matthias.hauswirth@usi.ch

**Silvia Santini**

Silvia Santini is since September 2016 an Associate Professor at the Faculty of Informatics of USI.

From July 2014 until August 2016, Silvia held an Associate Professor position at TU Dresden, where she led the Embedded Systems Lab. From October 2011 until July 2014 she was an Assistant Professor at the Department of Electrical Engineering and Information Technology of TU Darmstadt, Germany, where she led the Wireless Sensor Networks Lab. Previously, from 2009 until 2011, she was a postdoctoral researcher in Prof. Friedemann Mattern's Distributed Systems Group at the Department of Computer Science of ETH Zurich, Switzerland. From November 2010 until February 2011 she joined Leonidas Guibas's research group at Stanford University as a visiting scholar. Silvia completed her PhD Thesis under the supervision of Prof. Friedemann Mattern at ETH Zurich in 2009 and graduated in Telecommunication Engineering (with honors) from the Sapienza University of Rome, Italy, in May 2004.

**Piotr Didyk**

Piotr Didyk is an assistant professor at Università della Svizzera italiana (USI) where he leads a group on Perception, Display, and Fabrication. He received a Ph.D. in computer science from Saarland University and Max Planck Institute for Informatics in 2012. Later, he spent two years at Massachusetts Institute of Technology as a postdoctoral associate. In 2014, he became an independent research group leader at Cluster of Excellence Multimodal Computing and Interaction, Saarland University. At that time, he was also appointed as a Senior Researcher at Max Planck Institute for Informatics. He moved to Lugano in spring 2018. His research interests include human perception, computer graphics, new display technologies, image/video processing, 3D printing, and computational fabrication. His primary goal is to combine a deep understanding of human perception with computation and hardware to improve display quality as well as 3D fabrication techniques.  
piotr.didyk@usi.ch

**Michael Multerer**

Michael Multerer is a Mathematician who works on uncertainty quantification for partial differential equations. He is interested in developing and implementing efficient algorithms to address the resulting high dimensional problems. Michael received his PhD in Mathematics at the University of Basel in 2014. Afterwards, he had several postdoc positions at the University of Basel, the EPF Lausanne and the ETH Zurich. In 2018, he started as assistant professor at the Institute of Computational Science at the USI Lugano.  
michael.multerer@usi.ch

**Andrea E. Rizzoli**

Andrea E. Rizzoli holds a Ph.D. in Control Engineering and Informatics awarded from Politecnico di Milano (Italy). He is a senior research scientist at IDSIA and a professor at SUPSI. His interests are in simulation and in decision support systems and he has been involved in the development of various simulation models in a number of different contexts, from natural resources modelling to logistics and industrial systems. He has authored more than 100 publications on these subjects. In 2001 he has been awarded the 'Early Career Research Excellence' award by the Modelling and Simulation Society of Australia and New Zealand and in 2004 he became a Fellow of the International Environmental Modelling and Software Society, whose aim is to develop and use modelling and software tools to advance the science and improve decision making with respect to resource and environmental issues. He serves editor for the international journal Environmental Modelling & Software. In 2012 he has been elected President of the International Environmental Modelling and Software Society (<http://www.iemss.org>) for the biennium 2012-2014.  
andrea-emilio.rizzoli@usi.ch

**Robert Soulé**

Robert Soulé is an adjunct professor at the Università della Svizzera italiana (USI). His research interests are in distributed systems, networking,

and applied programming languages. Prior to joining USI, he was a postdoctoral associate at Cornell University. He received his PhD from New York University in 2012, and his BA from Brown University in 1999. For two years, he was a research co-op in the Data Intensive Systems and Analytics Group at IBM T. J. Watson Research Center. He is the recipient of an IBM Invention Plateau Award, a Google Faculty Research Award, and Best Paper Awards at ACM DEBS 2012 and USENIX NSDI 2018.  
robert.soule@usi.ch

## Teaching post-docs and researchers

Marco D'Ambros  
Alberto Ferrante  
Monica Landoni  
Andrea Mocchi  
Simone Pezzuto  
Mauro Prevostini  
Andrea Rosà

## External Lecturers

Massimo Banzi  
Marco Brambilla  
Cinzia Capiello  
Anthony Cleve  
Giovanni Denaro  
Alessandro Giusti  
Kazuki Irie  
Vittorio Limongelli  
Alessandro Lomi  
Davide Martinenghi  
Antonietta Mira  
Pierluigi Plebani  
Paolo Rossetti  
Alexander Stivala  
Michael Wand

---

122	Publisher	USI Università della Svizzera italiana
	Production	Faculty of Informatics
	Grid design	CCRZ
	Layout	Servizio grafica
	Number of copies	160
	Date	September 2021
	Orders	USI Università della Svizzera italiana Via Buffi 13 6900 Lugano, Switzerland +41 58 666 4000 +41 58 666 4647 info@usi.ch www.usi.ch

Università  
della  
Svizzera  
italiana



Faculty  
of  
Informatics

Plan of studies  
3-5-8

**2021/22**