Teaching plan of the Computer Science Area

Academic Year 2024/2025

The main teaching activities of the course in Computer Science are allocated in the first year; students have to obtain 60 CFUs (after Crediti Formativi Universitari, corresponding to the European Credit Transfer and Accumulation System) to progress to the second year. The taught part of the course —worth 47 CFUs— is divided in three terms scheduled as follows:

- Term 1: Immigration (from November 4 to December 12 2024; total CFUs 15)
- Term 2: Core (from Junuary 13 to April 17 2025; total CFUs 20)
- Term 3: Advanced (from May 5 to July 31 2025; total CFUs 12)

The remaining 13 CFUs are awarded if the thesis proposal is approved by the academic board.

Immigration

The modules in the first term are designed to provide students with basic and general knowledge in Computer Science with a focus on algorithms, formal methods, and software engineering —the three main research themes of the Computer Science group.

The modules, consisting of 30 hours of lectures worth 5 CFUs, are as follows:

- Design and Analysis of Algorithms (Lecturers: A. Balliu, M. Flammini, and D. Olivetti)
- Introduction to Formal Methods (Lecturers: F. Raimondi, C. Trubiani, and E. Tuosto)
- Introduction to Software Engineering (Lecturers: F. Basciani, M. De Sanctis, Di Salle, L. Iovino, and P. Pelliccione)

Moreover, students also attend the 10-hours long module

- How to (Lecturer: C. Trubiani)

about transferrable skills in research. This module does not award CFUs.

Core

In the second term students are offered a number of 14-hours long modules, each one worth 4 CFUs. The core modules are:

Algorithms

1. Algorithmic Game Theory (Lecturer: M. Flammini)
2. Approximation Algorithms (Lecturer: G. D'Angelo)
3. Graph Mining (Lecturer: P. Crescenzi)
4. Distributed Graph Algorithms (Lecturers: A. Balliu, D. Olivetti)

Formal Methods

- Formal Methods at Work (Lecturer: C. Trubiani)
- Introduction to Blockchain and Smart Contracts (Lecturer: M. Murgia)
- Model Checking (Lecturer: E. Tuosto)
- Satisfiability Problems and Applications (Lecturer: F. Raimondi)

Software Engineering

- Autonomous and Self-Adaptive Systems (Lecturers: M. De Sanctis, P. Pelliccione)
- Continuous Evolution of Software and Artifacts (Lecturers: A. Di Salle, L. Iovino & P. Pelliccione)
- Engineering of Human-centric and Smart Systems (Lecturers: M. De Sanctis, P. Inverardi, P. Pelliccione, G. L. Scoccia, N. Troquard)
- Software Testing (Lecturer: A. Bertolino)

- Theory and Applications of Model-Driven Engineering (Lecturers: F. Basciani, A. Di Salle, L. Iovino)

Besides the modules above, designed to introduce students to background topics relevant to the research developed by the Computer Science group, students are offered interpillar modules designed to provide cross-cutting technical skill applicable in more than one pillar. These modules are

- Introduction to Python Programming and Machine Learning (Lecturers: ???)
- Knowledge Representation and Ontology Engineering (Lecturer: N. Troquard)

The modules above are optional; students must attend (at least) five modules of their choice for which they have to pass the corresponding exam. Students can attend more than five modules.

Advanced

Advanced modules span specific topics for which there an active ongoing research activity in the scientific community. Each module is worth 1 CFU and consists of 6-10 hours of lectures. These modules are meant to provide students with a broad view of current research topics in Computer Science.

1. . . . (Lecturer: . . .)

Several other modules are delivered by guest lectures, expert of the topics. Such modues will be scheduled in due course. Students must attend at least 12 advanced modules.

1. Advanced Technologies for the Analysis of Cyber Physical Systems (Lecturer: S. Ali)
2. Casual Limits of Distributed Quantum Computation (Lecturer: F. D'Amore)
3. Fair Division: Elicitation, Algorithms and Justice Criteria (Lecturer: Y. Zick)
4. Foundation of Machine Learning (Lecturer: E. Natale)
5. Modelling and Validation of Concurrent Systems (Lecturer: A. Ravara)
6. Normative Requirements for Autonomous Systems (Lecturer: S. G. Yaman)
7. Ontologies and Robotic Applications (Lecturer: Stefano Borgo)
8. Protocol Specifications for Concurrent and Distributed Systems (Lecturer: N. Yoshida)
9. Quantum Computing Demystified (Lecturer: M. Tschaikowski)
10. Software Engineering for Cognitive Robots and Systems (Lecturer: N. Hochgeschwender)
11. Software Engineering with and for AI (Lecturer: F. Palomba)
12. Temporal Graphs: A Primer (Lecturer: A. Marino, A. Silva)
13. Testing, Debugging and Program Repairing (Lecturer: L. Mariani)
14. Geospatial Data Analysis (Lecturer: Maurizio Napolitano)

wherer the last module is an interdisciplinary module offered in collaboration with the Social Sciences group.

Learning Outcomes

After attending an advanced module, a student should

- be aware of the main results and open problems of the area
- be familiar with the main techniques used in a given domain
- have an awareness of the significance of the main results of the area

Summary

A summary of the teaching allocation is reported in the table below:

Module	Hours	Term	CFUs	Lecturer(s)
Design and Analysis of Algorithms	30	1	5	Balliu, Flammini, Olivetti
Introduction to Formal Methods	30	1	5	Raimondi, Trubiani, Tuosto
Introduction to Software Engineering	30	1	5	De Sanctis, Iovino, Pelliccione
How To	10	3	0	Trubiani
Algorithmic Game Theory	14	2	4	Flammini
Approximation Algorithms	14	2	4	D'Angelo
Graph Mining	14	2	4	Crescenzi
Distributed Graph Algorithms	14	2	4	Balliu, Olivetti
Formal Methods at Work	14	2	4	Trubiani

Module	Hours	Term	CFUs	Lecturer(s)
Introduction to Blockchain and Smart Contracts	14	2	4	Murgia
Model Checking	14	2	4	Grabmayer, Tuosto
Satisfiability Problems and Applications	14	2	4	Raimondi
Autonomous and Self-Adaptive Systems	14	2	4	De Sanctis, Pelliccione
Continuous Evolution of Software and Artifacts	14	2	4	Di Salle, Iovino, Pelliccione
Engineering of Human-centric and Smart Systems	14	2	4	De Sanctis, Inverardi, Pelliccione, Scoccia, Troquard
Software Testing (Bertolino)	0	3	0	
Theory and Applications of Model-Driven Engineering	14	2	4	Basciani, Di Salle, Iovino
Introduction to Python Programming and Machine Learning (???)	0	3	0	
Ethics in Computer Science	14	2	4	Crescenzi
Knowledge Representation and Ontology Engineering	14	2	4	Troquard
Mining Software Repositories	6	3	1	Scoccia
Static Analysis for Software Security	6	3	1	Inverso
UX/UI: Fundamentals for Interface Design	6	3	1	Rebelo
Advanced Technologies for the Analysis of Cyber Physical Systems	0	3	0	
Casual Limits of Distributed Quantum Computation	0	3	0	
Fair Division: Elicitation, Algorithms and Justice Criteria	0	3	0	
Foundation of Machine Learning	0	3	0	
Modelling and Validation of Concurrent Systems	0	3	0	
Normative Requirements for Autonomous Systems	0	3	0	
Ontologies and Robotic Applications	0	3	0	
Protocol Specifications for Concurrent and Distributed Systems	0	3	0	
Quantum Computing Demystified	0	3	0	
Software Engineering for Cognitive Robots and Systems	0	3	0	
Software Engineering with and for AI	0	3	0	
Temporal Graphs: A Primer	0	3	0	
Testing - Debugging and Program Repairing	0	3	0	