

Module Description

Software foundations

General Information**Number of ECTS Credits**

3

Abbreviation

TSM_SoftFound

Version

2017.04.04

Responsible of module

Tiziano Leidi, SUPSI

Language

	Lausanne	Bern	Zürich	Lugano/Manno
Instruction	<input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E	<input checked="" type="checkbox"/> E
Documentation	<input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E	<input checked="" type="checkbox"/> E
Examination	<input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E	<input checked="" type="checkbox"/> E

Module category

- Fundamental theoretical principles
 Technical/scientific specialization module
 Context module

Lessons

- 2 lecture periods and 1 tutorial period per week
 2 lecture periods per week

Brief course description of module objectives and content

The goal of the module is to provide software-design competences covering the software development process, software modeling and object-oriented programming

Aims, content, methods

This course provides ground competences about contemporary software technologies to students at master level. The course will review the recent evolution of software engineering methodologies with focus on software architectures in the era of cloud computing and the internet of things.

Focus will be on consolidating the required transversal foundation of advanced technical skills for modern software development, with particular care on the role played by recent innovations, for example in functional programming and in distributed processing.

The course will be proposed as a combination of lectures and exercises, including practical demonstrations and laboratory development. Readings will be assigned to students as complementary deepening material.

The student will be provided knowledge about modern software design methodologies and software architectures, including:

- Elements of clean code.
- Design patterns in the era of functional programming.
- Software architecture for enterprise applications.
- The presentation model: theoretical and practical aspects.
- Cloud computing technologies.
- Software solutions for distributed data processing with map/reduce.
- Software architectures for stream processing.
- Computing for the IoT: from cloud computing to fog computing.
- The protocol stack in the IoT.
- Software technologies for cyber-physical systems.

Teaching and learning methods

Frontal theoretical lessons, demonstrations, examples, exercises

Prerequisites, previous knowledge, entrance competencies

Programming skills in Java, C and C++

Ground software engineering skills (in particular design pattern)

Literature

Assessment**Certification requirements for final examinations (conditions for attestation)****Written module examination**

Duration of exam: 120 minutes

Permissible aids: None