CS-305 Software engineering

Candea George

Cursus	Sem.	Type
Communication systems	BA5	Opt.
Computational science and Engineering	MA1, MA3	Opt.
Computer science minor	H	Opt.
Computer science	BA5	Obl.
HES - IN	H	Obl.

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

Summary

This course teaches the basics of modern software development, focusing on techniques and practices used to build computer software that meets high standards of quality, reliability, security, and maintainability.

Content

- Object-oriented design and reasoning
- Design patterns
- Principles of building reliable and secure software
- Testing and debugging
- Code layout and style
- Development processes
- Software project management
- Tools for writing, analyzing, and debugging code, as well as source code management

Being a good software engineer entails a continuous learning process. Unlike mathematics or physics, this field changes fast, thus making continuous and independent learning essential. This course prepares students to become lifelong auto-didacts that build upon the foundation of imutable principles governing good software engineering.

Keywords

software development, software engineering, software design, development processes, agile methods

Learning Prerequisites

Required courses

- Good Java programming skills
- CS-108 Practice of Object-Oriented Programming
- CS-210 Functional Programming
- CS-206 Parallelism and concurrency
- CS-207 System-oriented Programming

Students who do not master the material taught in the prerequisite courses prior to starting Software Engineering typically do not manage to pass the course.

Recommended courses

The material in the following courses is helpful but not absolutely required:

- COM-208 Computer networks
- CS-208/209 Computer architecture

Important concepts to start the course

- Good knowledge of object-oriented programming (e.g., in Java)
- Knowledge of using version control systems (e.g., Git)

Learning Outcomes

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

- Design software that is reliable, secure, user-friendly, and performs well
- Implement (in software) sophisticated designs and algorithms
- Specify requirements for software systems
- Develop code that is maintainable
- Assess / Evaluate design and implementation options
- Choose alternatives to optimize for an objective

Transversal skills

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

Teaching methods

- Combination of online and in-class lectures
- Recitations and lab sessions
- Homework exercises

Expected student activities

- Attend and participate in lectures and recitations
- Watch online lectures
- Read and understand assigned materials
- Complete homework assignments independently

Assessment methods

Throughout the semester (contrôle continu). The final grade will be determined based on two exams during the semester and potentially quizzes administered during the semester. Exact formula may vary from year to year.

Supervision

Office hours	Yes
Assistants	Yes

Forum	Yes
Others	See http://sweng.epfl.ch/

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

See http://sweng.epfl.ch for up-to-date bibliography

Websites

- http://sweng.epfl.ch/