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	Written
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: designing software, working in a team, writing good code, shipping software, and evolving software. It emphasizes building software that meets high standards of quality, reliability, security, and manageability.

Content

Writing software

- Modularity
- Interfaces
- Software architecture

Getting software right

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

Shipping software

- Development processes
- DevOps
- Software evolution

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

Keywords

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

Learning Prerequisites

Required courses

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

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

Important concepts to start the course

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

Learning Outcomes

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

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

Teaching methods

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

Expected student activities

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

Assessment methods

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

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Bibliography

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

Ressources en bibliothèque

-
-

Websites

- https://sweng.epfl.ch