CIVIL-127

Programming and software development for engineers

Alahi Alexandre, Menghrajani Alok Deshmukh

Cursus	Sem.	Type
Civil Engineering	BA2	Obl.

Language of teaching	English
Coefficient	3
Session	Summer
Semester	Spring
Exam	During the semester
Workload	90h
Weeks	14
Hours	**3 weekly**
Lecture	1 weekly
Practical work	2 weekly
Number of positions	

Summary

This is a python programming course to build on top of students' existing programming skills and help write better software. The course will teach best practices, and introduce refactoring, fixing bugs, and implementing features in a large code base.

Content

- First 6 weeks
  - Focus on Python (any feature available in 3.12.0 (latest stable) is in scope)
    - Basic data types (str, int, float, bool, list, tuple, sets, dictionaries)
    - Loops
    - Control flows
    - Objects
    - Writing complete programs
    - Tooling (IDE, Jupyter, etc.)

- Remaining 8 weeks
  - Writing testable code / automated testing
    - Unittesting
    - Integration testing
    - Dependency Injection
    - Code coverage
    - Test driven development
  - Debugging techniques
    - E.g., Julia Evan's debugging manifesto
  - Software design ideas
    - Patterns (gang of four)

- Shallow vs deep classes
- Refactoring techniques and pitfalls (multi-cursor, sed, codemod, semantic patch, etc.)

- git
  - Understanding the core logic (DAG)
  - Learning basic commands for the cli
  - Git bisect (with a linear history)
  - Git bisect with non-linear history
  - Splitting large commits into smaller chunks

- Github
  - How to fork a project
  - How to track issues
  - How to open a PR
  - Github Actions (including CI/CD)

- Towards verification of program correctness
  - Linters
  - Type systems
  - Proof systems

- Reproducibility
  - Reproducible builds using Docker
  - Reproducible experiments

- Cloud tools
  - Discuss tools students are likely to use on AWS, GCP, or other clouds.

- Technical writing
  - Writing technical documents
  - Writing cleaner and more effective error messages

For the exercices, given a large piece of code, the student will perform some refactoring tasks + implement new features. This would be similar to a new hire's experience when joining a software engineering team.

Keywords

Python, software development

Learning Prerequisites

Required courses

Basic knowledge of python

Learning Outcomes

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

- Apply industry-standard best practices in Python programming.
- Design , build and maintain large software projects effectively.
- Develop clean, testable and easily reviewable code.
- Assess / Evaluate existing codebases to improve code quality and functionality.
- Test , debug and fix bugs in a systematic and efficient manner.
- Implement new features in existing large-scale software projects.

Transversal skills

- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Respect relevant legal guidelines and ethical codes for the profession.

Teaching methods

Class in-person lectures + labs

Expected student activities

Attend to classs lectures/labs, complete weekly programming assignements and project.

Assessment methods

MCQ exam + grades programming assignements

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Moodle Link

- https://go.epfl.ch/CIVIL-127