BIO-210

Applied software engineering for life sciences

Mathis Alexander

Cursus	Sem.	Type
Life Sciences Engineering	BA3	Obl.

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

Summary

We learn and apply software engineering principles to program projects in Python. Projects cover problems in life sciences, and will be developed over the course of the semester.

Content

- Python (object types, statements, functions, packages, object oriented programming)
- Distributed version control via git
- Debugging, profiling, refactoring
- Unit, integration and functional testing
- Project and code documentation
- Models of developmental biology and neuroscience

The first part of the semester is devoted to acquiring the necessary skills and tools. In the second part, biological problems are presented and the students form groups of 2 to 3 members to create a python repository that addresses one biological question. The software has to meet various specifications with regard to the application programming interface, documentation and performance. The hours of practical work will be devoted to planning, coding and presenting.

Keywords

Python, software engineering, pattern formation, associative memory

Learning Prerequisites

Required courses

Object-oriented programming and Information, Computation, Communication

Learning Outcomes

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

- Design an application meeting given specifications
- Optimize the adequacy of a program in relation to the targeted functionalities

- Use tools dedicated to the realization of software projects (version control, debugging, profiling)
- Develop a medium-sized application using python
- Use known libraries and interface other programming languages
- Interpret software documentation

Transversal skills

- Write a scientific or technical report.
- Evaluate one's own performance in the team, receive and respond appropriately to feedback.
- Resolve conflicts in ways that are productive for the task and the people concerned.
- Manage priorities.
- Use both general and domain specific IT resources and tools
- Keep appropriate documentation for group meetings.
- Assess progress against the plan, and adapt the plan as appropriate.
- Set objectives and design an action plan to reach those objectives.

Teaching methods

Lectures with code examples, practical work on computers, problem sets, and realization of one graded project.

Expected student activities

Participation in the course. Realization of problem sets and projects in exercise sessions and individual work during the week.

Assessment methods

The final mark is a combination of 3 evaluations: individual work for the problem sets in the first classes (40%), evaluation of the second project carried out as a team (35%), individual contribution to teamwork (25%).
Each team will be graded by assessing unit tests, profiling and project results. The individual grade will be based on participation and contributions to the repository. The evaluation criteria for final projects take into account:

- full history on git
- integration and completeness of tests
- quality of documentation, clarity of code
- code quality, program performance, elegance of visualizations
- understanding algorithms

Supervision

Office hours	Yes
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Moodle Link

- https://go.epfl.ch/BIO-210