Scientific Software Development with Python

Simon Pfreunschuh (Course responsible), Donal Murtagh (Examiner)

August 17, 2020

Scientific Software Development in Python is a graduate course covering the topics **software development, project management and scientific computing**. It aims to provide participants with the necessary knowledge to develop high-quality, reusable and efficient scientific software as part of a larger team or on their own.

1 Course plan

The course consists of two parts:

1. Lecture series (2.5 ECTS): The lecture series will take part during autumn term 2020 (September - December) and cover the following topics.

    - Software development:
        - Project planning and management
        - Organizing collaboration
        - DevOps with Python and GitHub: Version control, continuous integration, testing
    - General programming principles:
        - Object oriented design
        - Computing models and architectures
    - The Python programming language
        - Python language basics
        - Standard library
        - Package management
        - Testing and documentation of python packages

- Scientific computing:
    - Parallel and distributed programming
    - Out-of-core programming
    - Visualization
    - Distributed and cloud computing

2. Programming project (5 ECTS): The project part of the course will consist of a programming project that participants will collaborate on in groups of at least two. Participants are encouraged to choose a their project according to their research interests. The project comprises three phases:

    - Planning phase: Participants will together define the scope and use cases for the project.
    - Implementation phase: Participants will work on implementing the project. There will be regular progress meeting and code reviews.
    - Final phase: Each participant will present her results and insights in a seminar.

During the project there will be regular retrospectives to reflect on project progress and lessons learned.

2 Intended learning outcome

After the course, participants will be able to:

- Write, package and deploy high-quality Python software
- Know how and when to apply different computing techniques
- Contribute to (open source) software projects

3 Time plan

3.1 Lectures

3.1.1 Part 1: Software development and programming principles

- Week 1: Introductory lecture

  - Aims of software development

- Week 2: Project management and planning
  - DevOps and continuous integration
  - Agile software development

- Week 3: Programming principles
  - Procedural vs. object oriented programming
  - Basics of object oriented design

- Week 4: 1st retrospective:
  - Progress report from project groups
  - Leaning and open challenges

3.1.2 Part 2: The Python programming language

- Week 5: Overview of the Python programming language
  - Python programming principles
  - Language features
  - The Python package system

- Week 6: The Python standard library

- Week 7: Python programming recipes

- Week 8: 2nd retrospective:
  - Progress report from project groups
  - Learnings and open challenges

3.1.3 Part 3: Scientific programming with Python

- Week 9: Parallel programming with Python:
  - The Python Interpreter Lock
  - Python co-routines

- Week 10: Distributed programming with Python
  - IPyParallel

    - MPI
    - Out of core programming

- Week 11 (Optional): Scientific visualizations with Python
- Week 12: 3rd retrospective:
    - Progress report from project groups
    - Learnings and open challenges

3.2 Projects

3.3 Week 1-4: Planning phase

- Definition of project scope
- Setup of the development environment
- Outlining a project plan and first prototyping
- Last week: Retrospective and refactoring
    - Defining aim and scope of 1st implementation phase

3.4 Week 5-8: 1st Implementation phase

- Implementation of the project code
- Last week: Retrospective and refactoring
    - Defining aim and scope of 2nd implementation phase

3.5 Week 9-12: 2nd Implementation phase

- Further work on project code
- Last week: Retrospective and refactoring
- Presentations from project members.