PhD Course Syllabus

Practical Application of Python in Engineering and Scientific Research

1 Course Description and Objectives

This course aims to teach students the practical use of the Python language for solving engineering and research problems through script creation. The curriculum is designed to guide students from analyzing problems and designing algorithmic solutions to implementing them using both basic and advanced Python structures.

Upon completion, students will be able to efficiently and independently manage Python versions and packages, load and process data from various file formats, and perform advanced data analysis. Students will also acquire the skills to create their own tools and functions, organize code into coherent structures, and automate workflows by running external software. The course emphasizes the use of universal development tools (such as VS Code and git) and key scientific libraries.

2 Prerequisites

- No specific programming prerequisites are listed, as the course covers environment setup and basic concepts from the ground up.
- Familiarity with basic computer operations and file systems is recommended.

3 Course Contents

1. Development Environment
   - Introduction to the *conda* environment as a Python package manager.
   - Introduction to the Visual Studio Code (VS Code) Integrated Development Environment (IDE).
   - Installing the conda package with Python.
   - Introduction to the Jupyter environment.
   - Introduction to local and remote work workflows.
2. Python fundamentals
   - Introduction to the basics of variables and operations on Python variables.
   - Basic syntax and data types.
3. Advanced Programming Concepts
   - Introduction to creating functions, classes, and methods.
   - Object-oriented programming foundations.
   - Organizing code into larger, coherent structures.
4. Scientific Computing and Data Analysis
   - Workshops on the scientific use of packages such as `numpy`, `pandas`, `scipy`, and `xarray`.
   - Analyzing and processing loaded data.
5. System Interaction and Automation

    - Workshops on using packages to interact with processes and files: `os`, `sys`, `Pathlib`, `subprocess`, `multiprocessing`.
    - Running external software and managing files/folders.

6. Data Processing and Storage
    - Workshops on using the regex package for advanced text operations.
    - Workshops on packages for writing data to files: `json`, `h5py`, and others.

7. Performance Optimization
    - Workshops on packages for accelerating Python performance: `numba`, `dask`, and others.

4 Hours and Schedule

- Total Hours: approximately 15 hours (Based on the sum of listed modules) .
- ECTS Points 1.5
- Schedule: The course consists of 5 lectures/workshops, each lasting 3 hours.

5 Grading Policy

The course is conducted entirely as practical workshops and exercises. Students are expected to solve practical exercises throughout the course to demonstrate acquired competencies, such as managing environments, processing data, and automating tasks.