Engineering Tripos Part IA, 1P4: Computing, 2024-25

Lecturer

Dr F Mancini [1]

Timing and Structure

Michaelmas Term: week 1, 1 introductory lecture; weeks 1-, 12 independent exercises: Lent Term: week 1, 1 lecture; weeks 1-, group project

Prerequisites

None

Aims

The aims of the course are to:

- Introduce students to computing for engineering applications.
- Introduction to programming in Python.
- Enable students to devise and implement algorithms to compute solutions to problems.
- Develop foundational software engineering skills.
- Develop skills for team-based software development, including use of version control.

Objectives

As specific objectives, by the end of the course students should be able to:

- Describe using text and mathematics the purpose and flow of a program.
- Write and run Python progams in (i) Jupyter notebooks and (ii) from multiple source files.
- Understand variables, assignment, simple operators and precedence.
- Appreciate the importance of types and the pitfalls of round-off error and floating point arithmetic.
- Use of data structures and libraries.
- Understand the concept of an algorithm and algorithmic complexity.
- Apply error handling and unit testing as part of good software engineering practice.
- Develop skills for numerical computing for engineering applications.
- Be able to develop simple object-oriented data structures.
- Fetch data from different sources, and manipulate the data and display graphically.

Content

Michaelmas Term

The Michaelmas Term part of the course involves 12 activities for self-study, and each activity has exercises to be completed. The exercises for *at least* the first six activities must be competed by the middle of the Term and will be checked at a sign-up session, and the remainder must be competed by the sign-up session at the end of the Term.

- Familiarisation with the Jupyter environment for Python, including use of LaTeX for displaying mathematics

- Variables and assignment of values
- Control statements (if, for and while)
- Types and floating point arithmetic
- Functions
- Libraries
- Numerical computation, including array processing
- Data plotting
- Code testing and error handling
- Algorithms
- Complexity
- Data structures
- Object oriented design

Lent Term

The Lent Term activity is a group exercise, with students working in pairs. Each student takes charge of writing part of a software solution. A modular design and unit testing are required to ensure that the two parts work together correctly.

- Problem solving using abstraction and modularisation
- Structured programming and program modularisation using functions
- Using data structures
- Using library functions and handling exceptions
- Developing and running programs written in multiple source files
- Use of git for version control

Further notes

There are separate web pages associated with each Term's coursework:

https://github.com/CambridgeEngineering/PartIA-Computing-Michaelmas [2]

https://cued-partia-flood-warning.rtfd.io/ [3]

Examples papers

There are two examples papers: the first one is issued over the Christmas vacation, the second over the Easter vacation.

Booklists

Please refer to the Booklist for Part IA Courses for references to this module, this can be found on the associated Moodle course.

Examination Guidelines

Please refer to Form & conduct of the examinations [4].

Last modified: 30/07/2024 08:44

ia-1p4-computing-2024-25

Links
[1] mailto:fm456@cam.ac.uk
[2] https://github.com/CambridgeEngineering/PartIA-Computing-Michaelmas
[3] https://cued-partia-flood-warning.rtfd.io/
[4] https://teaching24-25.eng.cam.ac.uk/content/form-conduct-examinations