Engineering Tripos Part IA, 1P4: Computing, 2017-18

Course leader

Dr Garth N. Wells [1]

Lab leader

Tim Love [2]

Timing and Structure

Michaelmas Term: week 1, 1 introductory lecture; weeks 2-, 12 independent exercises: Lent Term: week 1, 1 lecture; weeks 2-, group exercise

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 good 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://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Mich... [3]

http://cued-partia-flood-warning.rtfd.io/ [4]

Examples papers

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

Booklists

Please see the Booklist for Part IA Courses [5] for module references.

Examination Guidelines

Please refer to Form & conduct of the examinations [6].

Last modified: 12/12/2017 06:43

Source URL (modified on 12-12-17): https://teaching17-18.eng.cam.ac.uk/content/engineering-tripos-part-ia-1p4-computing-2017-18

Links
[1] mailto:gnw20@cam.ac.uk
[2] mailto:tpl@eng.cam.ac.uk
[3] https://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Michaelmas
[4] http://cued-partia-flood-warning.rtfd.io/
[5] https://www.vle.cam.ac.uk/mod/book/view.php?id=364071&chapterid=42031
[6] https://teaching17-18.eng.cam.ac.uk/content/form-conduct-examinations