

00 Part IA Michaelmas Term computing

September 13, 2020

1 Part IA Michaelmas Term computing

This is the introductory notebook for Michaelmas Term computing at the Department of Engineering at University of Cambridge.

The course is for first year students in their first term, and does not assume any prior computing experience. It does assume a level of mathematical knowledge.

2 Structure

Computing is best learned by doing, and seeking help when you need it. This course involves 12 self-study, interactive Jupyter notebook ‘activities’, with exercises at the end of each activity. The activity notebooks cover:

1. [Variables, assignment and operator precedence](#)
2. [Control statements](#)
3. [Types, type conversions and floating point arithmetic](#)
4. [Functions](#)
5. [Library functions](#)
6. [Data structures](#)
7. [Numerical computation](#)
8. [Plotting](#)
9. [Error handling and testing](#)
10. [Algorithms](#)
11. [Complexity](#)
12. [Object-oriented design](#)

The notebooks are designed to be completed in order. To complete an activity:

1. Read through a notebook, and change parts of the programs if you want to experiment with what happens - this will help your understanding.
2. Complete the exercises for the notebook before moving onto the next activity.

As you move through the activities you might need to revise some of the earlier notebooks.

3 Completing exercises

Exercises for each notebook are in the [Exercises](#) directory.

- Use Markdown cells, and where appropriate add equations using LaTeX, to describe what your program is doing.
- For longer exercises, you may want to break your program into multiple cells, with descriptions in Markdown before cells with code.
- Most exercises include some automated testing to provide feedback on the correctness of your implementations. You are not expected to understand the test syntax to start.

You are required to present your exercise solutions at a marking session. Check the lab rota for your timetabled marking slot.

4 Getting help

The best resource is the Internet. Cambridge engineering can visit the Help Desk in the DPO

4.1 Providing help

Peer support is encouraged - if you receive help make sure that you do understand what you have done.

5 Jupyter notebooks

This is a Jupyter Notebook. We will be computing using Jupyter Notebooks (<https://jupyter.org/>) and the programming language Python (<https://www.python.org/>). Jupyter Notebooks provide an interactive environment where you can mix text, equations, programming and visual outputs. This technology is increasingly widely used, and it is all open-source. You will find Jupyter Notebooks useful throughout the course.

5.1 Using notebooks

See the notebook [Using notebooks](#) for guidance.

5.2 Running notebooks

5.2.1 Azure notebook service

Using the Jupyter service hosted at <https://notebooks.azure.com/> is recommended. Log in using your CRSid.

5.2.2 Running locally

Jupyter can be run locally. If you choose to do so, the [Anaconda](#) distribution is recommended. Be sure to download the Python 3 version. Anaconda is also installed in the DPO.

5.3 Python version

We will be using exclusively Python 3. There are some subtle differences between Python 2 and Python 3. Bear this in mind if using a book or searching online.

6 Other resources

There are many online resources for learning computing, Python, and on using Jupyter notebooks. A search engine is the best way to find relevant material. Remember that we are using Python 3 and not Python 2 - you might want to add 'python 3' to any search query.

A useful resource for beginners is <https://www.codecademy.com/learn/learn-python-3>.