Introduction to Computer Programming

1.1 Course Introduction

University of BRISTOL

Why study programming?

- Increased use of computing in everyday life.
- A tool you can use for the other subject you study.
- A growing sector of the jobs market (software engineering, data science, AI, robotics...).
- Coding in jobs, not traditionally related to computing.
- It's fun!

Why study Python?

- Free and open source
- "High level" (abstracted from computing processes eg. memory management)
- Easy to learn
- Versatile - both a scripting and a programming langauge:
  - scripting : run in a host appication for debugging and viewing output e.g. a mathematical model
  - programming : controls a computer or machine e.g. a microcontroller on a ro
- Increasingly used in industry
- Community

Course Goals

- A good standalone programming toolkit.
- Skills to improve the quality of your work in other subjects.
- A fundamental base from which to start developing further as a programmer.

Course Entry Level

Beginner, no prior programming knowlegde.
Preparation for first year (and upwards), engineering / engineering-related degree programmme

Course stucture: Weekly schedule

Lectures

	Group 1	Group 2	Online students
2 hour lecture/lab	Tuesday	Tuesday	Monday
	13:00-15:00	15:00-17:00	8:00-10:00
	MVB 2.11	MVB 2.11	remo

Optional drop-in sessions

	On-campus	Online
Optional drop-in support session	Friday	Thursday
	13:00-15:00	9:00-10:00
	MVB 1.15	remo

Group 1 : EMAT, Innovation, Biorobotics, everyone not listed in Groups 1/2

Group 2: EENG, EDES, Digital Health MSc & CDT

Class Structure

Lecturers: Hemma Philamore, Matthew Hennessy

Part 1 (1 hour)

- 15-30 mins theory/demos
- 30-45 mins practise exercises (essential + advanced questions)

Part 2 (1 hour)

- 15-30 mins theory/demos
- 30-45 mins practise exercises (essential + advanced questions)

Complete any unfinished 'essential' exercises for homework.

How To Access the Course Material

Blackboard page for unit EMAT10007:

- Slides (pdf and Jupyter notebook)
- Weekly exercises (pdf)

- Example answers released the following week

Different ways of running Python

We wil run Python code (scripts) in different environments:

- Jupyter notebook (lecture slides)
- Spyder IDE (intergated development environment)

Write, debug and see the output of our code.

Learning Python **syntax** which can be run in many other environments (computer command line, raspberry pi etc...)

In [5]:

```
# A cell for writing and running code
```

Installing Anaconda

Instructions for installing Anaconda on your personal computer:

- Windows : https://docs.anaconda.com/anaconda/install/windows/ (https://docs.anaconda.com/anaconda/install/windows/)
- Mac : https://docs.anaconda.com/anaconda/install/mac-os/ (https://docs.anaconda.com/anaconda/install/mac-os/)
- Linux : https://docs.anaconda.com/anaconda/install/linux/ (https://docs.anaconda.com/anaconda/install/linux/)

Opening Anaconda

Linux lab

- Open Terminal from programs menu
- `/opt/anaconda/2020.07/bin/anaconda-navigator`
- Press enter

Personal computer

- Choose Anaconda from the programs menu

Lecture notres can be viewed as pdf or Jupyter notebook

Opening Jupyter notebook

Click 'Launch' next to the Jupyter notebook application

Opening lecture notes in Jupyter notebook

Navigate to where you have downloaded the .ipynb file to and click on it to open.

Dependencies

The folders `sample_data` and `img` should also be downloaded and stored in the same folder as the .ipynb files.

Opening Spyder

Click 'Launch' next to the Spyder application

How to save your work in Spyder

You will complete the weekly exercises in spyder.

These are effectively your notes.

Save them in an organised way (e.g using the week/class number).

File >> Save >> [`Filename`] (.py file extension automatically added)

How you will be asssed

Assignment: Single piece of coursework, completed individually.

Theme: Write a program to perform and encryption and decryption task + a short (2 page) report

Set: Week 12 (December 13th 2021)

Deadline:

Any Questions?

Let's give it a go!...

In [ ]: