Introduction to Computer Programming

1.1 Course Introduction

University of BRISTOL

What is programming?

Programming is a way to tell a computer to do a specific task e.g.

- adding two numbers
- plotting some data as a graph
- saving a file

Like, humans. computers understand instructions given in particular (programming) languages.

Computers understand instructions that are written in a specific *syntactical* form called a programming language.

Why study programming?

- A tool you can use for the other subject you study.
- Increased use of computing in everyday life.
- 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 application for debugging and viewing output e.g. a mathematical model
    - programming : controls a computer or machine e.g. a microcontroller on a robot
- 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.
First year (and upwards), engineering / engineering-related degree programmme

Course stucture: Weekly schedule

2 hour Lecture/lab

Group 1	Group 2	Online students
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
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:

2 hour lecture/lab

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 (not assessed).

Please sit in the same seat/area of the lab each week - TAs (TA profile picture (remo) / lanyard (on-campus) will monitor the same areas of the lab.

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

Access these files during the class using a web browser.

Different ways of running Python / software we will use

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

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

IDE (integrated development environment)

Allows you to do the following within a single piece of software:

- write and edit code
- run code and see the output
- debug code (graphical dispay showing th anture nad cause of an error)

In [10]:

```python
B = 4

print(B)
```

```
4
```

IDE: allows us to write, debug and see the output of our code.

We will learn Python **syntax** .

Python may be be run in many other environments (computer command line, raspberry pi etc...)

Installing Anaconda

Anaconda installs a number of applications including:

- Jupyter notebook
- Spyder IDE

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/)

(You should have already installed Anaconda if attending the online class)

Opening Anaconda

You may use the lab computers or personal laptop computers in the lecture:

Linux lab

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

Personal computer

- Choose Anaconda from the programs menu

ANACONDA®

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 from blackboard and click on it to open.

You can now edit code slides e.g. copy in-class examples

Alternativey, open as pdf e.g. in web browser

Dependencies

If viewing slides using Jupyter notebook...

The folders `sample_data` , and `img` should be stored in the *same folder* as the lecture slides (.ipynb files) for the images and code examples to appear correctly in the lecture slides.

It is recommended that you store **all slides**, **`sample_data`** , **`img`** together in a single folder.

You can download `sample_data` , and `img` from the main blackboard page.

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: End of November

Deadline: Before Christmas break

UoB covid mitigation guidelines

- Stay at home if you have symptoms
- Wash your hands
- Take a Lateral Flow Device test twice a week
- Wear your face covering when on campus
- Observe local measures to keep you safe

https://www.bristol.ac.uk/students/your-studies/study-2021/health-safety/ (https://www.bristol.ac.uk/students/your-studies/study-2021/health-safety/)

Any Questions?

Let's give it a go!...