

U30299 – Programming

Introduction to Module

Matthew Poole & Nadim Bakhshov
`moodle.port.ac.uk`

School of Computing
University of Portsmouth

2020/21

Introduction to module

- The aim of this module is for you to develop your programming skills, either:
 - from scratch, or
 - building upon your current skills.

Programming

- A program is a set of instructions that performs a **task**.
- We can interpret (or carry out) sets of instructions in English; e.g:

put pen on paper
move pen left 10cm
move pen down 20cm
move pen right 10cm
take pen off paper
move pen up 10cm
move pen left 10cm
put pen on paper
move pen right 5cm

- What task does this “program” perform?

Programming

- We are interested in **computer programs**: sets of instructions to be carried out by a computer.
- Programming is the process of constructing computer programs.
- For a computer to carry out (or execute) our programs, we have to write them in a **programming language**.
- For the first teaching block, we'll write (quite small) programs in the (relatively simple) **Python** programming language.
- Later, we'll learn **object-oriented programming**, an approach suitable for writing larger programs.
- For this, we'll use Python and a second language, **Java**.

Developing your programming skills

- Programming is a **skill** (like painting or playing tennis).
- Like these other skills, programming:
 - can **only** be developed properly through **practice**;
 - should be fun!
- Knowing how to program well is important:
 - later during your course, and
 - for many of you, during your career.
- This module aims to give you the:
 - basic knowledge; and
 - guidance/helpto enable you to develop your programming skills (by practising!).

Module structure and organisation

- The module includes:
 - **learning materials** (lecture notes, videos, worksheets) on Moodle;
 - **timetabled sessions** (tutorials, practicals and drop-ins).
- Each week, you'll find on Moodle:
 - lecture notes for **two lectures** in PDF format;
 - **videos** that complement the lecture notes.
- These lectures cover programming knowledge and techniques that serve as a foundation for you attempt the worksheets.
- You should aim to study each week's lectures before your **tutorial class of that week**, where we will discuss and test your knowledge of the topics covered.
- A weekly **worksheet** will also be put on Moodle; you should aim to complete the worksheet before your **practical class of the following week**.

Tutorial classes

- Each week you will attend a tutorial class. Your timetable will show both an on-campus and an online session each week (you will attend only one of these).
- The on-campus tutorial will be on your timetable at one of the following times: Tue 3:30pm, Wed 10am, Wed 11am or Thur 9am.
- Since room capacities are very limited, you will **need to register** for each on-campus tutorial; details will be sent by email at the beginning of each week.
- Although labelled on the timetable as a “drop-in”, you need to arrive by the start time, and tutorials will last around 45mins.
- If you do not attend the on-campus tutorial (e.g., if it is fully booked or you need or wish to remain at home), then you should attend the online tutorial on Thur at 5pm.
- The online tutorial will be delivered through Zoom (the link will be on Moodle).
- Don't attend both the on-campus and online versions in the same week.

Practical classes

- Each week you will attend an online 1hr 45min practical session.
- This will be on your timetable at one of the following times: Mon 10am, Tue 1:30pm or Fri 9am.
- Practicals will typically have two parts:
 - an initial group part where you will work in small groups on an exercise;
 - an individual part where the staff will give you feedback on your efforts on the previous week's worksheet; this part will also allow you to request help from members of staff.
- The online practicals will be delivered through Zoom (the link will be on Moodle).

Learning Support Tutor for Computing

- Outside tutorials and practicals, there are other sources of help on the module.
- Xia Han is the Faculty of Technology's Learning Support Tutor for Computing (xia.han@port.ac.uk).
- You will be able to book one-to-one support sessions with Xia.

Online drop-in session

- Finally, if you would like to ask Matthew or Nadim for extra help outside the practical sessions, there is an online drop-in session on Thursdays 4pm-4:50pm.
- This an optional session for one-to-one support designed for a small number of students who would like some extra help or have a question about the module.
- This drop-in session will be delivered through Zoom (the link will be on Moodle).
- If you would like to make use of this session, please join before 4:30pm.

Studying outside class time

- Most of your time on this module will be spent improving your programming skills (practising) by working through the worksheets in your own time.
- You will also need to spend time studying the lecture materials before each week's tutorial.
- As a rough guide, you might dedicate 8 hours outside class time each week to this module.
- This may vary depending on your previous experience, the difficulty of the worksheet, etc.

Module staff

- Matthew Poole is module-coordinator, and organises the classes and assessments in Teaching Block 1.
- Email - `Matthew.Poole@port.ac.uk`
- Nadim Bakhshov organises the classes and assessments in Teaching Block 2.
- Email - `Nadim.Bakhshov@port.ac.uk`
- You may also see some of the following in your online practicals:
Alex Bennett, Peter Boyd, Mani Ghahremani, Soraya Harding, Simon Marsden and Alaa Mohasseb.

Topics covered in Teaching Block 1

week	tutorial review / worksheet handout
1	Writing simple programs
2	Computing with numbers
3	Graphics; writing quality code
4	Strings and files
5	Consolidation week - no lectures
6	Functions
7	Decision structures
8	Loop structures
9	Design and simulation, Lists (worksheet)
	Christmas break
10	Lists (tutorial)
11	No lectures planned

Assessment

- You will be assessed in several ways throughout the year, which will give you three final percentage marks:
 - **Item 1** – solutions to worksheet exercises, quizzes in tutorials and two end-of-teaching-block online computer-based tests (50% of module);
 - **Item 2** – a programming assignment using Python (25% of module); and
 - **Item 3** – a programming assignment using Java (25% of module).
- The weighted average from the marks you achieve in these items will give you a module mark—you need to achieve a module mark of 40% or more to pass.

Item 1 (50%) – Worksheets (20%), quizzes (10%) & tests (20%)

- In each week's practical class we will give you individual feedback on your attempt at the **worksheet** that was set (made available) during the previous week.
- As part of this feedback, we will give you a mark of between 0 and 3.
- You will have around 18 practical worksheets over the year; however:
 - the marks you get before the first consolidation week will not be counted;
 - to allow for some absence/illness etc., we will count only your best 10 marks from the remaining weeks' worksheets.
- Your 10 marks will be summed up and weighted 20% of the module (i.e. each worksheet will be weighted only 2% and each feedback mark will only count as $\frac{2}{3}\%$).
- It is important for you to focus on the feedback (in order to develop your skills for later assessments), rather than each very low weighted mark.

Item 1 (50%) – Worksheets (20%), quizzes (10%) & tests (20%)

- We will set many **quiz** questions (typically multiple choice) in the tutorial sessions.
- Like the practicals, you will have around 18 tutorials, but:
 - the marks you get before the first consolidation week will not be counted;
 - to allow for some absence/illness etc., we will count only your best 10 tutorials (those that you get the biggest proportion of questions correct) from the remaining weeks.
- You will need a web-enabled device (smartphone, tablet, laptop) in the tutorials for the quizzes.

Item 1 (50%) – Worksheets (20%), quizzes (10%) & tests (20%)

- You will also have two online **tests** of programming knowledge (with multiple choice and short-answer questions):
 - one in the assessment period in January on the topics covered in Teaching Block 1;
 - one in the assessment period in May/June on the topics covered in Teaching Block 2.
- These will each be weighted 10% of the module.

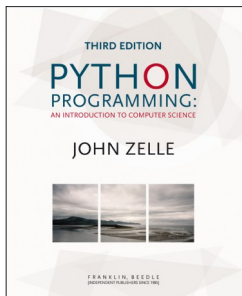
Item 2 (25%) and Item 3 (25%)

- The two programming assignments will be set for you to work on during the final few weeks of each teaching block (before the assessment periods):
 - Item 2 (Python) will be in Teaching Block 1;
 - Item 3 (Java) will be in Teaching Block 2.

Python software

- To write and execute our Python programs we'll need to use the Python software.
- It's available for free for your own machines at www.python.org.
- Make sure that you install and use a version that begins with '3' (version 3.8 is recommended).
- **Do not** use an old version that begins with '2'—our programs will work with it!
- We'll also use a free program editor for Python called Pyzo (www.pyzo.org).
- There are versions of Python and Pyzo for Windows, Mac or Linux.
- Please try to install Python and Pyzo before your first practical class.

Recommended Python book



- John Zelle
- Python Programming: an Introduction to Computer Science
- **3rd edition**
- Franklin, Beedle & Associates, 2016
- ISBN 9781590282755

- Currently around 30 – 45*GBP*, if you wish to purchase it.
- There are a few copies of the book in the library.
- **Do not** get the old 1st edition, but the 2nd edition should be fine.