2023/24 Module Handbook

**Computer**

**Programming**

School of Built Environment,

Engineering and Computing

Level 4

Semester A

(20 Credits)

CRN 12385



**Contents**

1 What this Module is About.................................................................................................1 1.1 Introduction from the Module Leader........................................................................1 1.2 Module Aims...............................................................................................................2 1.3 Module Learning Outcomes........................................................................................2 1.4 Module Learning Activities..........................................................................................2 1.5 Indicative Module Content .........................................................................................2 1.6 Graduate Attributes Developed and Assessed ...........................................................3 1.7 Communication...........................................................................................................3

2 Weekly Schedule ................................................................................................................4 3 Key Resources to Support Learning....................................................................................5 4 Assessment.........................................................................................................................5

4.1 Assessment Summary .................................................................................................5

4.1.1 Assessment 1: Phase Test...............................................................................................6 4.1.2 Assessment 2: Portfolio ..................................................................................................6 4.1.3 Reassessment..................................................................................................................6 4.2 Assessment Details......................................................................................................6

4.3 Feedback on Your Assessments..................................................................................6 5 Understanding Your Assessment Responsibilities .............................................................7 5.1 Extenuating Circumstances and Mitigation ................................................................7 5.2 Late Submission...........................................................................................................7 5.3 Academic Misconduct.................................................................................................7 6 The End ...............................................................................................................................8

**Student Name** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Email Address** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Course** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Group** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Module Tutor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Tutor’s Email Address** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Communication Protocol:*** *module staff will reply to student questions within a reasonable time but this will normally be within office hours only. Students are advised to check this handbook and also to see if there are any online/noticeboard announcements or FAQ answers that deal with their enquiry before contacting staff.*

**1 What this Module is About**

**1.1 Introduction from the Module Leader**

Greetings!

Welcome to the University and in particular welcome to this introduction to Computer Programming.

Computer Programming is one of the most important aspects of computing. After all, if we can’t make the computer do something useful, why are we here at all? Without a program a computer is just a pile of components that make a sort of humming noise when powered on. Useful for a cat to sleep on, maybe, but not useful for doing the accounts.

Whatever your plans are for the future, you *need* to understand the basics of programming. Maybe you plan to become a professional developer (big skills shortage there), using the skills you will start to learn here to solve business problems, produce new computer games, or do other cool stuff. Or maybe management is more your thing – in that case you’ll need to know enough programming so that the developers don’t mess you round! Or forensics – you won’t want to be checking every sector on that drive without some help…

We are going to learn using a language called Python, but what we are learning is *programming*. In later modules you’ll meet other programming languages – PHP, Java, maybe more – which you should be able to pick up quickly if your Python is good.

Remember above all that you learn to program with practice. I learned to program at University many years ago, but I *really* learned to program when I did it for a job. Knowing that my code was running on a test rig testing diesel engines that would explode if I’d fouled up kind of focuses the mind.

So:

Practice. Then practice more. Then practice a bit more.

and you’ll get there.

The rest of this handbook contains useful information on how the module will run. Have a read and bring any questions to your first practical session.

Cheers!

*Tony*

Module Leader: Tony Jenkins

Email: a.m.jenkins@leedsbeckett.ac.uk

2023/24 Module Handbook 1

**1.2 Module Aims**

This module aims to introduce fundamental programming concepts, specifically focusing on procedural programming and general-purpose input and output techniques. Students will also be introduced to the use of a modern interactive development environment, and to the basics of source code control.

**1.3 Module Learning Outcomes**

By studying this module, the learners are expected to:

• Develop an understanding of the key concepts of procedural programming, including sequence, selection, and iteration, along with the use of variables and data types. • Have the ability to write functioning computer programs written using a third-generation programming language, which take input, perform processing, and then generate output. • Gain an understanding of the typical software tools used to carry out computer programming activities, and have the ability to use these tools to develop, debug and execute program code.

The assessments provide the opportunity for students to demonstrate that they have attained the outcomes.

**1.4 Module Learning Activities**

Tutorials introducing the module content will be available on MyBeckett. These will include video lectures, along with shorter videos focusing on key topics. The collection will be added to as the module progresses, for example in response to questions.

Students should view this content before their weekly practical session. In the practical session there will be a range of activities intended to reinforce the theory and to develop skills across the development life cycle.

Each week concludes with an online lecture summarising the main business of the week. This will be conducted on Microsoft Teams. Synchronous attendance is encouraged, but is optional. A recording of the lecture will be posted on MyBeckett.

In addition, students will need to allocate some of their own time each week to self-guided study. Links to resources to support this (online tutorials, books, and the like) will be found on MyBeckett. Students who find other useful resources are encouraged to add to the collection!

**1.5 Indicative Module Content**

The aim in the module is to cover the basics of Python programming. This includes concepts common to all programming languages, as well as some specific to Python itself. Python will be used in a “pythonic” way, demonstrating best practice with this particular language. That said, examples may be used from other languages to illustrate the common concepts that exist.

2023/24 Module Handbook 2

The module will begin with discussion of basic programming concepts such as sequence and selection, along with variables and values. These will then be extended with more complex data structures, and input/output concepts as complete programs are built. The final weeks will include some more advanced topics as pointers towards later modules.

Students will also be introduced to the use of a modern interactive development environment and will be encouraged to start to understand how they individually work best. Some tools will be recommended, but the overall approach will be platform-agnostic in terms of IDE and operating system.

Basic source code control will also be introduced as a way of making sample code available. **1.6 Graduate Attributes Developed and Assessed**

Graduate attributes provide a language for students to articulate their skills and strengths, and a basis to self-assess their personal development. The graduate attributes in this module are developed and assessed:

| ***Attribute*** | ***How Developed*** | ***How Assessed*** |
| --- | --- | --- |
| **Enterprise** | Problem solving skills are  developed, and placed in the wider context of solving real world business problems. | By demonstrating completed code and good practice in  program development. |
| **Digital Literacy** | Many digital literacy capabilities are used and developed in  investigating the module content. | Successful development of tested programs. |
| **Global Outlook** | Students are developing skills that are globally recognised and which meet a significant skills shortage in industry. | Awareness of the bigger picture in systems development. |

**1.7 Communication**

The main formal means of communication will be via MyBeckett and your University student email account. You should ensure that you check your University email account every other day. Staff will not respond to emails sent from any other email account.

For short questions you are encouraged to use the “Chat” feature of Microsoft Teams rather than email. This is much more efficient, and you will get an answer sooner. If asking about a program error be sure to provide the offending code as well as the error message.

If you feel you are falling behind it is important to contact the Module Team as soon as possible. In this module in particular it is very difficult to catch up if you do fall behind.

If you are going to apply for mitigation you will need to provide written evidence of the reason for your absence (see Section 5 for further information).

2023/24 Module Handbook 3

**2 “Weekly” Schedule**

The general plan is shown below. Like all good plans it is subject to change. This plan is indicative, and does not necessarily correspond to weekly events. It is in the nature of this module that some students will progress faster than others; this is fine as long as the end point is the same for all.

|  | **Topic** | **Overview** | **Activities** |
| --- | --- | --- | --- |
| **1** | *Introduction* | Context; Types of Programming Language; Current Trends;  Python; IDEs; Simple Expressions. | Using the interpreter; Starting the IDE; Basic expressions; Writing “Hello World”. |
| **2** | *Variables and Types* | Variables; Assignment and Data Types; Output; Strings; Simple Compound Data Types. | Working with variables; identifying data types; using functions;  indexes and slicing. |
| **3** | *Control Statements* | Booleans; Boolean Expressions; Decision Making; Membership Testing; Iteration. | Using the “if” statement; using the “in” operator; using “while” and “for” loops. |
| **4** | *Functions* | Importing Functions; Defining Functions; Function Parameters; Lamda Functions. | Using the “def” keyword; default and keyword parameters; variable numbers of parameters. |
| **5** | *Scripts and Modules* | Running Programs in Files;  Command-line Arguments; Creating Modules. | Executing programs outside the IDE; writing portable programs; using command-line arguments. |
| **6** | *Lists and Tuples* | Compound Data Types; Lists and Operations; Tuples; Mutability and Immutability. | Using lists; writing list  comprehensions; packing and unpacking. |
| **7** | *Sets and Dictionaries* | Sets; Key-Value Pairs and  Dictionaries; Review of  Compound Data Types. | Using sets; using dictionaries; applications of dictionaries;  comprehensions; iterations. |
| **8** | *File Handling* | Files; Reading from Files; Writing to Files; Common File Formats: CSV and JSON. | Opening files; using read() and write() operations; closing files; modules for common formats. |
| **9** | *Errors and Exceptions* | Types of Error; Approaches to Error Checking; Handling Runtime Errors; Logging. | LBYL and EAFP approaches;  checking conditions; exception handling; defensive programming. |
| **10** | *Conclusions* | Review; Current Trends in  Programming; Futures; Skills Development. |  |

In addition, use of different development tools and basic source code control operations with Git will be covered in the practical sessions throughout the module.

2023/24 Module Handbook

4

**3 Key Resources to Support Learning**

Resources for the module will be linked from MyBeckett.

There are many books on Python and programming, but as with all printed media they struggle to keep up with changes in the technology. A small number of recommended texts will be listed on MyBeckett.

Online tutorials for Python abound on the web, and a search will turn up many excellent free resources. There are also many tutorials available on YouTube. A curated list will be maintained on MyBeckett.

Students wishing to work on their own machine will need to download and install Python. This is free and can be found via www.python.org. Users of Windows can also install Python via the “Microsoft Store”. Users of Linux will probably find that it just works, and is already installed. The module will use Python 3 (at the time of writing the current version is 3.11, but anything after 3.9 will be fine). Students should be very sure they have installed Python 3, as some of the module content will not work under Python 2.

Students are free to use any IDE they want. We recommend JetBrains PyCharm, which is available free of charge under a student licence. See:

https://www.jetbrains.com/community/education/#students

Many alternatives exist. Microsoft Visual Studio is a popular alternative and is equally free.

Students may, of course, use any text editor along with a command-line. Visual Studio Code (a free download from https://code.visualstudio.com/ ) is free, popular, integrates well with source code control, and is available for most platforms. Linux users need look no further than “vi”.

Students are strongly encouraged to keep their code under version control, using Git with a remote repository. This will make moving work between home and campus seamless. GitHub is recommended, but GitLab or BitBucket are also free. The University also provides a Git facility, but this is available only on campus, or via a VPN.

*Code kept online should obviously be in a secure private repository.*

Any students requiring additional support or alternative arrangements must declare this and provide evidence of their disability to the Disability Advice Team as early as possible.

**4 Assessment**

**4.1 Assessment Summary**

There are two assessments. An online test carries 40% of the marks, with the remaining 60% coming from a repository of source code.

Detailed specifications are in the Assessment folder on MyBeckett. Basic details are below.

2023/24 Module Handbook 5

4.1.1 Assessment 1: Phase Test

| **Assessment Method:** | Online Test | **Re-assessment**  **Method:** | As original. |
| --- | --- | --- | --- |
| Length | 75 Minutes | Length | As original. |
| Assessment Date | During Week 7 | Re-assessment Date and Time: | April 2024, precise date to be confirmed. |
| Feedback Method: | Immediate, via VLE. | Feedback Method: | As original. |
| Feedback Date: | Immediate. | Feedback Date: | Immediate. |

4.1.2 Assessment 2: Portfolio

| **Assessment Method:** | Repository | **Re-assessment**  **Method:** | As original. |
| --- | --- | --- | --- |
| Assessment Date | Repo to show no  commits after 24th  December 2023. | Re-assessment Date and Time: | April 2024, precise date to be confirmed. |
| Feedback Method: | Feedback and marking grid via VLE. | Feedback Method: | Feedback and marking grid via VLE. |
| Feedback Date: | 3 working weeks | Feedback Date: | Results release. |

4.1.3 Reassessment

The usual rules for reassessment apply. In both cases the reassessment work is to enhance and correct the shortcomings of the work originally submitted. You will be given feedback on the original submission, along with a marking grid that clearly shows where improvements can be made.

**4.2 Assessment Details**

The assessments are provided in the “Assessment” folder on the VLE. The version found there should be considered definitive. To avoid confusion it is not reproduced here.

**4.3 Feedback on Your Assessments**

Feedback forms a large part of your learning experience and is vital to your personal and professional development.

You will get written feedback on both assignments, which will be sent to you via MyBeckett. Formative feedback will also be available before submission.

Feedback on how well you are doing in your studies is important to your learning. The methods of feedback used in this module are:

2023/24 Module Handbook 6

• Formative feedback during online sessions, by email and by Teams: when you ask a question and receive an answer or when your tutor comments upon your work or drafts of submissions.

• Summative feedback: written feedback with the returned assessments and verbal feedback providing general comments about your work or the work of the class.

You are encouraged to provide feedback about the module to the module and course team.

Regular reviews will be included in the module. These are opportunities to resolve module issues promptly before they become too significant. In addition, you will have the opportunity to complete a module evaluation at the end of your module. These comments will be reviewed by the whole course team and may be considered at course planning meetings.

**5 Understanding Your Assessment Responsibilities**

**5.1 Extenuating Circumstances and Mitigation**

If you are experiencing problems which are adversely affecting your ability to study (called “extenuating circumstances”), then you can apply for mitigation. You can find full details of how to apply for mitigation at: www.leedsbeckett.ac.uk/studenthub/mitigation.htm.

The University operates a fit to sit/fit to submit approach to extenuating circumstances which means students who take their assessment are declaring themselves fit to do so. More information is available at the above link and here: http://www.leedsbeckett.ac.uk/studenthub/examinations/

**5.2 Late Submission**

Without any form of extenuating circumstances, standard penalties apply for late submission of assessed work. Full details of the penalties for late submission of course work are available at www.leedsbeckett.ac.uk/public-information/academic-regulations.

If you anticipate that you will submit work late, it is *always* best to discuss this with the module leader beforehand.

**5.3 Academic Misconduct**

Academic misconduct occurs when you yourself have not done the work that you submit. It may include cheating, plagiarism, self-plagiarism, collusion and other forms of unfair practice. What is and what is not permitted is clearly explained in *Factsheet for Students* which is available to view at: www.leedsbeckett.ac.uk/studenthub/academic-integrity.

There are a range of resources available to help you understand what is and what is not permitted and how to use other people’s ideas in your assessed work. These include the Skills for Learning website which can be found at http://skillsforlearning.leedsbeckett.ac.uk

The serious consequences of plagiarism and other types of unfair practice are detailed in the Academic Regulations at www.leedsbeckett.ac.uk/public-information/academic-regulations.

2023/24 Module Handbook 7

If you find that you are tempted to cheat, for example by submitting programs written by another, you are strongly encouraged not to, and to instead discuss how you might produce your own work. This tends to lead to a better outcome for everyone.

It is simple in programming to tell if the person presenting a program is the actual author, and a demonstration where they are not can become very uncomfortable. Even if a student did “get away with it” they are only storing up problems for the future in later programming modules.

**6 The End**

Hopefully, this document has explained all you need to know about the module. If not, do not be afraid to get in touch.

Enjoy the module!

This version: 07/09/2023 10:26

2023/24 Module Handbook 8