UNIVERSITY OF GREENWICH

MODULE HANDBOOK

COMP1811 – Paradigms of Programming

2023 – 2024

Contents

1 Welcome message from your Module Leader

Programming is fundamentally the conceptual foundation for the study of computation. The science of computing can be thought of as the implementation of logic and critical thinking. Learning to programme effectively is core to your success in the industry. This module is, therefore, a prerequisite for almost every other module in the Computer Science and Computing related programmes at the University of Greenwich.

The module introduces computer programming using different programming paradigms, which include functional and object-oriented programming. You will gain an understanding of the key commonalities, differences, and trade-offs between these paradigms and their applicability to different programming problems. Through practical coding exercises, you will develop key design, problem-solving, and coding skills that emphasise quality of software design for scalability and reuse, and the need for a professional approach to software development.

Like any other skill, learning to programme takes plenty of practice to master. Through exposure to the different languages, you will build confidence in your ability to learn and take on new programming languages - the aim is to "learn how to learn" new languages. As a polyglot software developer, you will broaden your employability prospects in a constantly evolving information technology industry with rapidly changing requirements.

This handbook provides essential information about this module including the aims and learning outcomes, the schedule of teaching and learning activities, assessment tasks, reading recommendations, and, if applicable, any additional resources that you will need. Please read it at the start of the term so you are aware of key details and important dates.

2 Key contacts (academic queries)

The list below provides contact details of the module team. Meetings with tutors during office hours will be arranged in their offices or via Microsoft Teams. Tutors will also be available on Moodle chat and email during office hours.

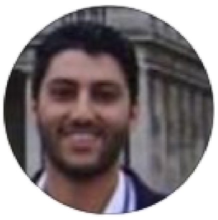
Note: Term 2 office hours are subject to change. Changes, if any, will be announced by your tutor at the beginning of Term 2.

Module Lead: **Dr. Yasmine Arafa**

Location: QM410
Email: Y.Arafa@gre.ac.uk
Tel: (020 8331) 7682
Office hours: (Term1 and 2)
Wednesdays 09:00 to 11:00
or by appointment

Module Tutor: **Dr. Mohammad Al-Antary**

Location: QM259
Email: M.AlAntary@gre.ac.uk
Tel: (020 8331) TBA
Office hours: (Term 1 and 2)
TBA



Module Tutor: **Dr. Ralph Barthel**

Location: QM362
Email: R.A.Barthel@gre.ac.uk
Tel: (020 8331) 7582
Office hours: (Term 1 and 2)
Wednesdays 11:00 to 13:00
or by appointment

Module Tutor: **Dr. Christopher Beckwith**

Location: TBA
Email: C.Beckwith@gre.ac.uk
Tel: (020 8331) TBA
Office hours: (Term 1 and 2)
TBA

Module Tutor: **Prof. Cornelia Boldyreff**

Location: QM420
Email: C.Boldyreff@gre.ac.uk
Tel: (020 8331) 8717
Office hours:
by appointment

Module Tutor: **Dr. Tuan Nguyen**

Location: TBA
Email: tuan.nguyen@gre.ac.uk
Tel: TBA
Office hours:
TBA

Module Tutor: **Dr.Pushparajah Rajaguru**

Location: TBA
Email: P.Rajaguru @ gre.ac.uk
Tel: TBA
Office hours:
TBA

Module Tutor: **Dr. Máté Szabó**

Location: TBA
Email: M.Szabo@gre.ac.uk
Tel: TBA
Office hours:
TBA

Module Tutor: **Dr. Rafael Torres**

Location: QM412
Email: R.MartinezTorres@gre.ac.uk
Tel: TBA
Office hours: (Term 1 and 2) Wednesdays 09:00 to 12:00 or by appointment

Module Tutor: **Dr. Andy Wicks**

Location: QM420
Email: Andrew.Wicks@gre.ac.uk
Tel: (020 8331) 8717
Office hours: (Term1 and 2)
Wednesdays 09:00 to 12:00
or by appointment

3 Module details and learning outcomes

Host Faculty: Faculty of Engineering and Science
Host School: School of Computing and Mathematical Sciences
Number of Credits: 30
Term(s) of delivery: 1 and 2
Site(s) of delivery: Greenwich Maritime Campus
Pre-requisite modules: None
Co-requisite modules: None

Aims:

To provide a solid foundation in programming concepts and hands-on experience using them. The module introduces computer programming using different paradigms, including object-oriented, functional and logical programming. You will gain an understanding of the key commonalities, differences and trade-offs between these paradigms and their applicability to different problems.

Learning Outcomes:

On successful completion of this module you will be able to:

1. Understand programming paradigms and their applicability to practical problems.
2. Apply appropriate programming constructs in each programming paradigm.
3. Design, implement and test small-scale applications in each paradigm.
4. Use appropriate tools to design, edit and debug program for each paradigm.Please insert the details below, taken from the current version of the Module Specification, as approved at the most recent approval or review event.

Glossary:

Aims	define the overall educational purpose of the module.
Co-requisite Module	is one that must be taken alongside this module.
Pre-requisite Module	is one that must have been completed successfully before taking this module.
Learning Outcome	is a subject-specific statement that defines the learning to be achieved through completing this module.

4 Employability

This module provides some of the essential basic programming skills needed for a career in the IT industry. In addition, it includes a number of employability skills that you will find useful when you embark on your career:

Knowledge

- Principles and application of the object-oriented and functional programming paradigms. In addition to the fundamentals of logical programming.
- Basic programming constructs and syntax of the Python and Scheme languages, as well as a brief introduction to Prolog.
- Key stages of software development including testing.

Technical Skills (essential practical skills for a career in computing)

- Apply the principles of object-oriented and functional program design in the development of software systems.
- Use an Interactive Development Environment appropriate for languages covered.
- Design, implement and test small software applications to conform to given system specification.
- Apply standard coding conventions and professional style.

Cognitive Skills

- Analytical skills

Coding involves identifying a problem and coming up with a technological solution to address it. You will acquire analytical skills that will enable you to understand the issue you're dealing with, evaluate different solutions, examine different options and select appropriate solutions to meet requirements.

- Problem solving and reflection

Research studies have shown that computer programming encourages cognitive development. Coding, in general, is often about evaluating large-scale, complex problems and breaking them down into smaller simpler pieces that can be tackled more easily. This is a much sought-after skill that readily transfers to many other areas of life and is highly valued by potential employers.

Generic Competencies (Skills for life and work)

- Teamwork: working as a member of a development team and recognising the different roles within a team and ways of managing and organising tasks.
- Self-management: managing workloads, and personal and professional development. Able to work unsupervised in an efficient, punctual, and structured manner to meet deadlines, and examine the outcomes of tasks and events, and judge levels of quality and importance.
- Perseverance: It is not uncommon for code not to work as expected on the first try! Programmers often take multiple attempts and a lot of work to get their programs to run smoothly. Quite often initial coding is scraped, and a completely different approach(es) is tried. You should be able to learn to handle such failure and to quickly evaluate alternative approaches.
- Technical report writing and documentation.

You can find out more about the Greenwich Employability Passport at: Greenwich Employability Passport for students. Information about the Career Centre is available at: Employability and Careers | University of Greenwich. You can also use LinkedIn Learning to gain access to thousands of expert-led courses to support your personal development. More information can be found at: LinkedIn learning | IT and library services

5 Schedule of teaching and learning activities

The teaching and learning material, and practice coding exercises are available on the Moodle pages for COMP1811. It is essential that you review all the teaching material each week and complete the self-assessment exercises and quizzes assigned.

***Note**: the schedule is subject to change and any changes will be announced accordingly*.

Week		Activity (lecture and lab topics)	Instructor
Term 1			
25/09/23	1.01 -	Introduction and Programming Basics	Y. Arafa
02/10/23	1.02 -	Python Basics: Variables, Data Types, and Operators	Y. Arafa
09/10/23	1.03 -	Data Structures and Sequences	R. Barthel
16/10/23	1.04 -	Flow Control: Conditionals and Boolean Logic	R. Barthel
23/10/23	1.05 -	Iteration and Repetition Structures	Y. Arafa
30/10/23	1.06 -	Organising & Reusing Code: Functions and Modules	Y. Arafa
06/11/23		Skills Week – No Lectures	
13/11/23	1.07 -	OO Concepts 1: Objects and Classes	R. Barthel
20/11/23	1.08 -	OO Concepts 2: Custom Data Types	R. Barthel
27/11/23	1.09 -	OO Concepts 3: OO Design - Identifying Classes	Y. Arafa
04/12/23	1.10 -	OO Concepts 4: Inheritance and Polymorphism	R. Barthel
11/12/23	1.11 -	Exception Handling, Testing and Debugging	M. Szabo
Term 2			
15/01/24	1.11 -	GUIs: Graphical User Interfaces	R. Barthel
22/01/24	1.12 -	Recursion	Y. Arafa
29/01/24	2.01 -	Functional Programming, Why? Scheme Introduction	Y. Arafa
05/02/24	2.02 -	A Gentle Introduction to Functional Programming	Y. Arafa
12/02/24	2.03 -	Recursion in Scheme	R. Torres
19/02/24	2.04 -	High-Order Programming	R. Torres
26/02/24		**Skills Week – No Lectures**	
04/03/24	2.05 -	Sides effects: state, sequencing, input/output	R. Torres
11/03/24	3.01 -	Logic Programming – Introduction, Concepts and Knowledge Representation	C.Boldyreff
18/03/24	3.02 -	Facts, Queries, and Rules	C.Boldyreff

Week	Activity (lecture and lab topics)		Instructor
25/03/24	3.03	Programming Paradigms in Python, Scheme and Prolog	C. Boldyreff
01/04/24	-	Bank Holiday	

*Includes discussion on coursework and another opportunity for you to ask questions.

In addition to the teaching and learning activities within the module, additional study support can be seen at: Academic Skills.

6 Attendance

Attendance and engagement with the module are really important! Programming is fundamental to the study of computation. Learning to program is essentially core to your success in the industry and on your program of study. It is important you attend all lectures and lab sessions and engage in the assigned activities every week. Face-to-face teaching and on-campus activities are important opportunities for you to connect with your lecturers and peers. Our previous students have told us how much they value in-person interaction with other students and the experience of an environment alive with academic scholarship and research.

You are expected to attend all timetabled face-to-face teaching sessions along with the labs. Attendance will be recorded for all in-person teaching sessions, including lectures, and labs. An entire week with no attendance at any module will normally be considered as a missed contact point.

Registering your attendance:

- In the lecture hall: The card readers will not be used due to the large number of students in each lecture session. You will have the option to scan the QR code shown on the screen as in the image below or go to a page in Greweb (available on the Greenwich Mobile App) and enter the One-Time Passcode (OTP). You will be asked to share your location (your location will only be shared at the point which you register your location only). It will not be possible to verify your location if you do not share your location.

[illegible]

Scan the QR code or use the passcode

Registration is open between [illegible]

Passcode: 398

Use in the Greenwich Mobile app or visit [illegible]

You will be asked to share your location. Your location will only be shared at the point you register your attendance. If you do not share your location it will not be possible to verify your attendance.

If you are unable to register your attendance, please speak to the staff member.

- In the IT Labs: Tap your ID card on the blue card reader located in the labs.

7 Assessment

Assessment schedule:

First sit assessments	Deadline	Weighting* out of100%	Mapped LO✦
CW1 – Practical Project (Python)	04/12/23	40	2, 3, 4
CW2 – Practical Project (Scheme)	18/03/24	30	2, 3, 4
Logbook	02/04/25	30	1, 2, 3

*The weighting refers to the proportion of the overall module result that each assessment task accounts for.
✦LO refers to learning outcomes.

Your assessment brief:

Coursework:

Both coursework are practical development projects that require the development of a system to given specifications using the object-oriented programming paradigm (Python) for CW1 and the functional programming paradigm (Scheme) for CW2. Additional application functionality may be announced during separate short hackathon sessions where you are required to submit additional functioning code. Each coursework requires a final report and an acceptance test (demonstration) of the final application. You are likely to fail the assessment if you do not attend the demonstration and show your code.

Start working on your coursework as soon as the specifications are released. You are advised to arrange frequent project meetings with your group/pair to distribute workload and follow-up on progress. Your lab tutor will periodically check your progress and ask you to demo the parts you complete at set milestones. This is an opportunity for you to receive formative feedback and support, which will help you improve your work.

Coursework Marking Criteria

a. Your code will be assessed on the following criteria:

- Features implemented and whether they are running correctly: The number of features you individually implemented from the list of requirements given in the coursework specification and their successful integration with code developed by others in your coursework group/pair.
- Additional feature implemented and running correctly (announced during a short hackathon session). Emphasis will also be on how well the additional feature integrates with the code produced for the main coursework and with the additional features implemented by others in your coursework group/pair.
- The quality of the code produced and the design. Credit will be given for the use of professional style: use of standard naming conventions, meaningful in-code comments, avoidance of code duplication, sensible naming, code indentation and structure, etc. The Python documentation pages provide more details. Scheme style will be discussed in the functional programming material.
- The user interface (where applicable). This is not a module about user interface design, but credit will be given for making your application as pleasant an experience as possible for the user.

b. Your report will be assessed on the following criteria:

- Are all the required sections included and completed properly?
- Does the report give an accurate reflection of what you have achieved?
- Is the report clear and easy read? Does it follow the structure specified?
- Is the evaluation realistic? does it show introspective thought about the work?

Logbook:

The logbook will include a number of tasks due at various stages in both terms 1 and 2 and is worth 30% of the overall grade of this module. Tasks are graded and may include coursework milestones, practical hackathon activities, and online quizzes. The quizzes are open-book tasks that are done during your scheduled lab session on campus and include multiple-choice and short coding questions covering all the topics taught on this module. The schedule for the logbook tasks will be outlined throughout the terms.

Important note: Coursework is marked on the understanding that it is your own work on the module and that it has not, in whole or part, been presented elsewhere for assessment. Where material has been used from other sources, this must be properly acknowledged in accordance with the University's Regulations regarding Academic Misconduct.

Marking, feedback and next steps

To pass this module, you achieve an overall mark of 40%+ for all assessments. For the coursework (coding projects) logbook, the marks and feedback will normally be provided within fifteen working days of the submission deadline. In exceptional circumstances, where there is a delay in providing feedback, you will be informed by the module leader.

If you do not pass a module in the first attempt, you will likely be eligible for a resit opportunity on the failed assessments. The Progression and Award Board (PAB) will decide whether you will be offered an opportunity to resit. **Note that marks on resit assessments are capped at 40% unless extenuation has been granted.**
For further details on resit assessments, please see section on Resit assessments below.
The assessment and feedback policy can be accessed at Assessment and Feedback Policy.

Moderation
Submitted student work is moderated by an academic moderator before the marks for an assessment are released. The purpose of moderation is to evaluate the overall standard of assessments on the module, and the quality of marking and feedback provided.

The moderator is unable to correspond with individual students about their work. If you need to discuss your marks and/or feedback, please contact your lab tutor in the first instance and the module leader if necessary.

Academic skills support
In addition to the teaching and learning activities within the module, additional academic skills support, guidance, and resources are available at the following links:

Academic and Digital Skills support - https://www.gre.ac.uk/academicskills
Academic Integrity - https://libguides.gre.ac.uk/courses/integrity
Guidance on use of AI - https://docs.gre.ac.uk/rep/information-and-library-services/ai-guidance
The IT handbook for new students - https://docs.gre.ac.uk/rep/information-and-library-services/student-booklet

Strong academic skills will help you to act with academic integrity, honesty, and trust. These are the values on which academic achievement at the University of Greenwich is based. As a student, you are expected to take responsibility for the integrity of your own work, including asking for clarification where necessary. Any improper activity or behaviour which may give you an academic advantage in assessment is considered to be assessment misconduct. Allegations of assessment misconduct will be considered under the University's Assessment Misconduct Procedure and may result in a penalty being imposed. More information about this procedure can be found at Assessment Misconduct Procedure.

Extenuating circumstances

The University recognises there are times when matters that are unexpected and beyond a student's control will impact on their performance and ability to complete assessments within the specified timeframe. Examples include unforeseen illness, a death in the family, or injury. Guidance on submitting an extenuation claim can be found at: Extenuating circumstances.

If you have a disability, specific learning difficulty, for example dyslexia, a long-term medical condition or a mental health condition which might affect your studies and assessments, and you have not already done so, then we advise that you seek support from the Student Wellbeing Service by contacting wellbeing@gre.ac.uk in the first instance

Student Support

The University offers a range of support services including health and medical services, a chaplaincy, disability and dyslexia support, and mental health & wellbeing support. Support can be accessed at Student Support | Support and Wellbeing.

8 Resit assessments

Assessment schedule:

Resit assessments	**Deadline**	**Weighting* out of100%**	**Mapped LO⁺**
CW1 - Python practical project	12/07/24	40	2, 3, 4
CW2 - Scheme practical project	12/07/24	30	2, 3, 4
Logbook	12/07/24	30	1, 2, 3

*The weighting refers to the proportion of the overall module result that each assessment task accounts for.
⁺LO refers to learning outcomes.

Assessment Brief:

Coursework:

Both coursework projects are to be implemented individually and require the development of applications to given specifications using the object-oriented programming paradigm (Python) for CW1 and the functional programming paradigm (Scheme) for CW2. Contact your lab tutor for support and assistance with resit coursework.

Coursework Marking Criteria

The same marking criteria outlined for the first sit coursework will apply to the resits apart from the group work element.

Logbook:

The resit logbook format is similar to the final exam and will consist of multiple tasks relating to all the topics covered on this module.

9 Resource recommendations

The following are suggested readings for the module. Additional, more detailed reading recommendations will be provided for the module topics.

You can check availability of the resources by using the search tool LibrarySearch at https://librarysearch.gre.ac.uk.

Author	Title	Publisher	ISBN
Hunt, John	A Beginners Guide to Python 3 Programming (2019). eCopy available in library.	Springer	978-3030202903
Beecher, Karl	Computational Thinking, A Beginners Guide to Problem-solving and Programming eCopy available in library. (2017)	BCS Learning Development	978-1780173658
Downey, Allen	Think Python: How to think like a Computer Scientist, 2nd Ed. (2015). eCopy available in library.	O'Reilly	978-1491939369
Matthes, Eric	Python crash course: a hands-on, project-based introduction to programming (2019) eCopy available in library.	No Starch Press	978-1593279295
Johnson, Mark	A Concise Introduction to Programming in Python, 2nd Ed. (2018)	CRC Press	978-1138082588
Friedman, Daniel & Felleisen, Matthias	The Little Schemer, 4th Edition (1995). eCopy available here.	The MIT Press	978-0262560993
R. Kent Dybvig	The Scheme Programming Language, 4th Edition (2009). eCopy available here	The MIT Press	978-0262512985
Michael Spivey	An introduction to logic programming through Prolog (2019). eCopy available here	Prentice–Hall International	

10 Digital Student Centre (non-academic queries)

Our new Digital Student Centre is your space to find answers 24/7 to your questions about student life, helping you get the support you need when you need it. AskUoG provides you with hundreds of up-to-date articles covering topics such as student engagement, student finance, academic and personal conduct, accommodation, visa and international student advice, disability, mental health and wellbeing support.
You can also download important documents like bank, student status and council tax letters by visiting My Documents on the Digital Student Centre (eligibility criteria apply).

If you can't find the right answer or need more personalised support for your query, you can create an enquiry and our specialist teams will respond swiftly. You will be able to track your requests and check the status of your enquiries in real time.

For academic queries, always contact the staff who work with you on your academic programme - your programme leader, module leader or personal tutor.

11 Changes to the module

At the University of Greenwich, we value feedback from students as well as External Examiners and other stakeholders and we use this information to help us improve our provision.

Important note: The University of Greenwich will do all that it reasonably can to deliver the module and support your learning as specified in our handbooks and other information provided. However, under some circumstances, changes may have to be made. This may include modifications to the:

- content and syllabus of modules, including in relation to placements
- timetable, location and number of classes
- content or method of delivery of your module
- timing and method of assessments.

This might be because of, for example:

- academic changes within subject areas
- the unanticipated departure or absence of members of university staff
- where the numbers expected on a module are so low that it is not possible to deliver an appropriate quality of education for students enrolled on it
- industrial action by university staff or third parties
- the acts of any government or local authority
- acts of terrorism.

In these circumstances, the university will take all reasonable steps to minimise disruption by making reasonable modifications. However, to the full extent that it is possible under the general law, the university excludes liability for any loss and/or damage suffered by any applicant or student due to these circumstances.