FOSE1030

Introduction to Python Programming

Session 1, In person-scheduled-weekday, North Ryde 2025

Science and Engineering Faculty level units

Contents

General Information

Unit convenor and teaching staff
Convenor, Lecturer
Matt Owers
matt.owers@mq.edu.au
Contact via email
By appointment.

Convenor, Lecturer
Michael Lay-Kujiraoka
michael.lay@mq.edu.au
Contact via email
By appointment.

Credit points
10

Prerequisites

Corequisites

Co-badged status

Unit description
Computer literacy has become a core skill across many disciplines. In this unit, students will learn coding in Python, enabling them to solve problems relevant to their discipline. They will leverage Python's foundational elements, including variables, operators, control structures, functions, and data structures, to develop effective programs. Students will adopt best practices in coding, such as a good code style and debugging capabilities, in an interactive practice-based environment. They will be introduced to the key Python libraries used for contemporary programming. The unit equips students to approach discipline-specific challenges with confidence, using Python as a versatile tool for problem-solving.

This unit has been designed for students requiring a Python background. For students enrolled in the Bachelor of Science, those in the Astronomy, Physics, Mathematics, and Statistical Data Science majors **must do this unit** and **NOT** FOSE1025.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

Learning Outcomes

On successful completion of this unit, you will be able to:

ULO1: Apply Python programming for effective problem-solving in and across disciplines.

ULO2: Utilize Python's fundamental elements (variables, operators, control structures, functions, and lists) for effective program development.

ULO3: Apply coding conventions to improve code readability and maintainability

ULO4: Employ debugging techniques to identify and fix common programming errors and issues.

ULO5: Utilize Python libraries and standard modules to accomplish diverse programming tasks.

General Assessment Information

Requirements to Pass the Unit

To pass the unit you must:

- Achieve a total mark greater than or equal to 50%

This unit does not contain any hurdle assessments.

Assessment Criteria

Assessment at Macquarie University is standards-based, as outlined in the Assessment Policy. This means that your work will be assessed against clear criteria, and these criteria will be made available to you on iLearn.

Submission of Assessments

All assessents must be submitted as descibed by the instructions provided on the iLearn page. Links for the submission of each assessment will be available on iLearn. ***Do not*** submit your assignments via email or in hard copy. The due dates for all assessment tasks are not negotiable. If you have commitments that will significantly impact your study during the session then you must plan for this in advance as part of an effective individual study plan and you may need to contact the unit convenor for advice.

Marking of Assessments

Assessment marks and feedback will be given via iLearn, with marks accessible via the Grades Report on iLearn.

We aim to return assessments marks and feedback within two to three weeks of the due date. We appreciate your patience and will advise you through iLearn when your marks and feedback are available for viewing.

Second Chances

For both the *Foundational Skills Assessment* and the *Final Exam*, students will be given an optional second chance to improve their grades. For both assessments, the second chance grade will be capped at a maximum of 84%.

Late Assessment Submission Penalty

Unless a Special Consideration request has been submitted and approved, a **5% penalty** (of the total possible mark of the task) will be applied for each day an assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is **11:55 pm**. A 1-hour grace period will be provided to students who experience a technical concern. For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration.

Assessments where Late Submissions will be accepted

- Foundational skills assessment - **NO**
- Programming assignment – **YES**, Standard Late Penalty applies
- Final Exam - **NO**

Special Consideration

The Special Consideration Policy aims to support students who have been impacted by short-term circumstances or events that are serious, unavoidable and significantly disruptive, and which may affect their performance in assessment. If you experience circumstances or events that affect your ability to complete the assessments in this unit on time, please inform the convenor and submit a Special Consideration request through https://connect.mq.edu.au.

Before submitting a special consideration, check whether you can make use of a second attempt offered to students regardless of a special consideration. This applies in particular to the *Foundational Skills Assessment* and the *Final Exam*. A special consideration should explain why the student missed both: the assessment task, and any offered second attempt.

Assessment Tasks

Name	Weighting	Hurdle	Due
Foundational skills assessment	30%	No	Week 8 during your registered SGTA.
Programming assignment	30%	No	06/06/2025
Final Exam.	40%	No	Week 13 during your registered SGTA.

Foundational skills assessment

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 24 hours
Due: **Week 8 during your registered SGTA.**
Weighting: **30%**

Students will be assessed on the foundational programming skills taught in the first half of the unit.

On successful completion you will be able to:

- Apply Python programming for effective problem-solving in and across disciplines.
- Utilize Python's fundamental elements (variables, operators, control structures, functions, and lists) for effective program development.

Programming assignment

Assessment Type [1]: Project
Indicative Time on Task [2]: 24 hours
Due: **06/06/2025**
Weighting: **30%**

Students will use the Python programming skills gained during SGTAs to solve an authentic, discipline-related problem.

On successful completion you will be able to:

- Apply Python programming for effective problem-solving in and across disciplines.
- Utilize Python's fundamental elements (variables, operators, control structures, functions, and lists) for effective program development.
- Apply coding conventions to improve code readability and maintainability
- Employ debugging techniques to identify and fix common programming errors and issues.
- Utilize Python libraries and standard modules to accomplish diverse programming tasks.

Final Exam.

Assessment Type [1]: Examination
Indicative Time on Task [2]: 24 hours

Due: **Week 13 during your registered SGTA.**
Weighting: **40%**

Students will be examined on content that was taught throughout the unit.

On successful completion you will be able to:

- Apply Python programming for effective problem-solving in and across disciplines.
- Utilize Python's fundamental elements (variables, operators, control structures, functions, and lists) for effective program development.
- Apply coding conventions to improve code readability and maintainability
- Employ debugging techniques to identify and fix common programming errors and issues.
- Utilize Python libraries and standard modules to accomplish diverse programming tasks.

---

[1] If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

[2] Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation

Delivery and Resources

Classes

Lectures (beginning in Week 1): There is one two-hour lecture each week. Lectures are very interactive and hands-on, and will involve coding demonstrations. For that reason, it is recommended that students **bring a laptop computer or tablet to the lectures each week.**

Small Group Teaching Activity (SGTA) classes (beginning in Week 1): Students should register in one two-hour SGTA class per week. Students should only attend the SGTA in which they have registered.

The timetable for classes can be found on the University website at: https://publish.mq.edu.au/. Enrolment can be managed using eStudent at: https://students.mq.edu.au/support/technology/systems/estudent

We strongly encourage all students to actively participate in all learning activities. Regular engagement is crucial for your success in this unit, as these activities provide opportunities to deepen your understanding of the material, collaborate with peers, and receive valuable

feedback from instructors, to assist in completing the unit assessments. Your active participation not only enhances your own learning experience but also contributes to a vibrant and dynamic learning environment

Unit communication

Unit staff will communicate with you via your university email or through announcements on iLearn. Queries to convenors should be placed on the iLearn General Forum.

For matters of a more personal nature, and that do not concern other students, you should contact the Unit Convener, Matt Owers, by email. Contact details are provided at the start of this document.

Technology Used and Required

Students will need to have access to a computer and internet to use the university systems (e.g. iLearn, library) and complete the assessment tasks.

We will make use of the following programming resources:

Python: https://www.python.org/

Visual Studio Code: https://code.visualstudio.com/

Online Python tutorials and compilors: https://www.tutorialspoint.com/

COVID Information

For the latest information on the University's response to COVID-19, please refer to the Coronavirus infection page on the Macquarie website: https://www.mq.edu.au/about/coronavirus-faqs. Remember to check this page regularly in case the information and requirements change during semester. If there are any changes to this unit in relation to COVID, these will be communicated via iLearn.

Computing Drop-in Centre (CDC)

FOSE1030 students may benefit from help at the Computing Drop-in Centre. The web page at https://students.mq.edu.au/study/faculties/science-and-engineering/drop-in-centre contains information, and further details will be provided on iLearn when available.

Unit Schedule

Week	Lecture material
1	Introduction and set up
2	Variables and operators
3	Conditions
4	Loops
5	Functions

Week	Lecture material
6	Lists
mid-sem	
7	Inbuilt Libraries
8	Numpy
9	Scipy
10	Plotting and Matplotlib
11	Pandas
12	Reading and writing files
13	Revision

N.B.: This schedule is flexible and subject to change.

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://policies.mq.edu.au). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy
- Fitness to Practice Procedure
- Assessment Procedure
- Complaints Resolution Procedure for Students and Members of the Public
- Special Consideration Policy

Students seeking more policy resources can visit Student Policies (https://students.mq.edu.au/support/study/policies). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

To find other policies relating to Teaching and Learning, visit Policy Central (https://policies.mq.edu.au) and use the search tool.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/admin/other-resources/student-conduct

Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit connect.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

Academic Integrity

At Macquarie, we believe academic integrity – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a range of resources and services to help you reach your potential, including free online writing and maths support, academic skills development and wellbeing consultations.

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

The Writing Centre

The Writing Centre provides resources to develop your English language proficiency, academic writing, and communication skills.

- Workshops
- Chat with a WriteWISE peer writing leader
- Access StudyWISE
- Upload an assignment to Studiosity
- Complete the Academic Integrity Module

The Library provides online and face to face support to help you find and use relevant information resources.

- Subject and Research Guides
- Ask a Librarian

Student Services and Support

Macquarie University offers a range of Student Support Services including:

- IT Support
- Accessibility and disability support with study
- Mental health support
- Safety support to respond to bullying, harassment, sexual harassment and sexual assault

- Social support including information about finances, tenancy and legal issues
- Student Advocacy provides independent advice on MQ policies, procedures, and processes

Student Enquiries

Got a question? Ask us via the Service Connect Portal, or contact Service Connect.

IT Help

For help with University computer systems and technology, visit http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/.

When using the University's IT, you must adhere to the Acceptable Use of IT Resources Policy. The policy applies to all who connect to the MQ network including students.

Changes from Previous Offering

This is a brand new unit and was added to the newly designed Bachelor of Science. The addition of this unit is a direct result of both student and staff feedback, which strongly indicated the need for a solid grounding in Python programming in order to enable success in upper-level units of Majors associated with numerical and analytical sciences.

We value student feedback to be able to continually improve the way we offer our units. As such we encourage students to provide constructive feedback via student surveys, to the teaching staff directly, or via the FSE Student Experience & Feedback link in the iLearn page.

Unit information based on version 2025.02 of the Handbook