COMP1000

Introduction to Computer Programming

Session 1, Online-scheduled-weekday 2023

School of Computing

Contents

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff
Convenor, Lecturer
Charanya Ramakrishnan
charanya.ramakrishnan@mq.edu.au
By appointment

Lecturer
Carl Svensson
carl.svensson@mq.edu.au
By appointment

Super Tutor
Samantha Kuhn
samantha.kuhn@mq.edu.au

Linden Misselbrook
linden.misselbrook@mq.edu.au

Credit points
10

Prerequisites

Corequisites

Co-badged status

Unit description
This unit is an introductory computer science unit, providing a practical introduction to basic computing and programming concepts. Students gain an understanding of, and practical experience in, computer programming; practical experience in implementing informal prose descriptions of problem solutions using an imperative language; an understanding of, and practical experience in, designing, coding, testing and debugging simple algorithms; and an understanding of the principle of incremental development. Other topics include: the concept of program correctness; the differences between high-level languages, assembly languages and machine languages; the role played by compilers; and the execution of programs by computer hardware.

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 problem solving skills to develop algorithms that solve small to medium-sized computational problems

ULO2: design and write code to implement a program description in an imperative programming language

ULO3: use standard software engineering practices to document, debug and test their programs

ULO4: understand and apply appropriately the concepts of variables, loops, functions, conditionals and compound data in the implementation of programmed systems

ULO5: identify and describe ethical issues in an academic environment and demonstrate active engagement in the learning process

General Assessment Information

Weekly SGTA Tasks

Every week in your registered SGTAs, you will be given tasks to work on. You will be submitting these tasks within the workshop hours to be eligible for these marks. Every week is worth 1% and you can score a maximum of 10%.

Programming Assessments

Out of the 40% for the programming assessments

- 10% is for assignment-1,
- 30% is for assignment-2 (15% for the evaluation of submitted work and 15% for the viva).
- Viva will be held in the registered workshops

Module exams

There are 3 module exams together worth 50%.

You must pass the module exams collectively (get 50% of the 50%, i.e 25/50 to pass the module exams). If you fail the exams, you will be given another attempt to pass the exams since this assessment is a hurdle. Failing this assessment means an automatic fail in the unit.

Late Submission for assessments: Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark) will be applied each day a programming 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 is provided to students who experience a technical concern.

Assessments where Late Submissions will be accepted

In this unit, late submissions will be accepted as follows:

- Programming Assessments: YES
- Module Exams: NO
- Weekly SGTA Tasks: NO

Special Consideration

If you cannot submit it on time because of illness or other circumstances, please apply for special consideration as soon as possible through https://ask.mq.edu.au/

For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, students need to submit an application for Special Consideration.

Assessment Tasks

Name	Weighting	Hurdle	Due
Weekly SGTA Tasks	10%	No	Weeks 1-12
Programming Assessments	40%	No	Weeks 6, 12, 13
Module Exams	50%	Yes	Weeks 3, 6, 11

Weekly SGTA Tasks

Assessment Type [1]: Practice-based task
Indicative Time on Task [2]: 10 hours
Due: **Weeks 1-12**
Weighting: **10%**

Weekly tasks during the SGTAs that students need to complete. Students must attend the SGTA and show their work to the tutor to be eligible for the mark.

On successful completion you will be able to:

- apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- design and write code to implement a program description in an imperative programming

language
- use standard software engineering practices to document, debug and test their programs
- understand and apply appropriately the concepts of variables, loops, functions, conditionals and compound data in the implementation of programmed systems
- identify and describe ethical issues in an academic environment and demonstrate active engagement in the learning process

Programming Assessments

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 20 hours
Due: **Weeks 6, 12, 13**
Weighting: **40%**

Multiple programming tasks spread through the semester where students put all their skills to work creating games or demos.

On successful completion you will be able to:

- apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- design and write code to implement a program description in an imperative programming language
- use standard software engineering practices to document, debug and test their programs
- understand and apply appropriately the concepts of variables, loops, functions, conditionals and compound data in the implementation of programmed systems
- identify and describe ethical issues in an academic environment and demonstrate active engagement in the learning process

Module Exams

Assessment Type [1]: Examination
Indicative Time on Task [2]: 42 hours
Due: **Weeks 3, 6, 11**
Weighting: **50%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

A number of exams spread through the semester. Students will have the opportunity to repeat any exam to improve their mark.

On successful completion you will be able to:

- apply problem solving skills to develop algorithms that solve small to medium-sized computational problems
- design and write code to implement a program description in an imperative programming language
- use standard software engineering practices to document, debug and test their programs
- understand and apply appropriately the concepts of variables, loops, functions, conditionals and compound data in the implementation of programmed systems
- identify and describe ethical issues in an academic environment and demonstrate active engagement in the learning process

[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

Each week you should attend

- two-hour lecture
- two-hour practical class

For details of days, times and rooms, consult the timetables webpage.

Note that Lectures and Practical classes commence in week 1.

You should have selected a practical class during enrolment. **You should attend the practical class in which you are enrolled**. You won't always get the class of your choice. Check availabilities via **eStudent** regularly. If ALL practical classes are full, only then, contact the convenor.

TEXTS AND/OR MATERIALS

Lecture notes: https://softwaretechnologymq.github.io/

Online tutorials on Processing website: https://processing.org/tutorials/

Learning Processing site (examples, exercises, videos): http://learningprocessing.com/

Textbook:

Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction, by Daniel Shiffman. Second edition, 2015. ISBN: 978-0123944436. **IMPORTANT:** Online/Digital version is available here on MQ Library. Click on "Elsevier ScienceDirect Books - Single Purchased Titles", log in using OneID credentials, and "Download all chapters".

Technology

Programming environment: Processing IDE

Flowchart generator: https://code2flow.com/

Web sequence diagram generator: https://www.websequencediagrams.com/

Diagrams: https://app.diagrams.net/ (lucid chart is better but this one is free)

Discussion Boards

The unit makes use of forums hosted within iLearn. Please post questions there, they are monitored by the unit staff.

Unit Schedule

1	Foundations of Programming	transition to processing, academic integrity module, study skills, how to use google	
2	Foundations of Programming	primitive operations, algorithms	
3	Variables & Conditionals	variables, debugging in processing, conditions	Module Exam-1 (5%)
4	Variables & Conditionals	variables, debugging in processing, conditions	
5	Loops	loops	
6	Loops		Module Exam-2 (20%)
7	Functions	functions	Assignment-1 (10%)
8	Functions	scope	
9	Compound Data	compound data	
10	Compound Data	reference semantics	
11	Program Design and Problem Solving	refactoring	Module Exam-3 (25%)

12	Program Design and Problem Solving	case study	Assignment-2 (15%)
13	Revision		Programming Assessment Vivas (15%)
Please note Every week from Week-1 to Week-12, you will have weekly tasks for submission in your workshop worth 1% each			

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 ask.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 an d maths support, academic skills development and wellbeing consultations.

Academic Integrity

Using the work or ideas of another person, whether intentionally or not, and presenting them as your own without clear acknowledgement of the source is called Plagiarism.

Macquarie University promotes awareness of information ethics through its Academic Integrity Policy. This means that:

- all academic work claimed as original must be the work of the person making the claim;
- all academic collaborations of any kind must be acknowledged;
- academic work must not be falsified in any way; and
- when the ideas of others are used, these ideas must be acknowledged appropriately.

All breaches of the Academic Integrity Policy are serious and penalties apply. Students should be aware that they may fail an assessment task, a unit or even be excluded from the University for breaching the Academic Integrity Policy.

Assessment Policy

Students should familiarise themselves with their responsibilities under the Assessment Policy, and notably the Final Examination Procedure.

Grade Appeals

A student who has been awarded a final grade for a unit has the right to appeal that grade as outlined in the Assessment Policy. Grade appeals apply to the final mark and the grade a student receives for a unit of study. They do not apply to results received for individual assessment tasks.

Grade appeals must be submitted via ask.mq.edu.au within 15 working days from the published result date for the relevant unit. Before submitting a Grade Appeal, please ensure that you read the Assessment Policy and note valid grounds for appeals.

Students are expected to seek feedback on individual assessment tasks prior to the award of a final grade. Students also have the right to request generic feedback from the teaching staff on their overall performance in the unit, including in a final examination. This can be done at any time in the six-month period starting from the day on which the final grade of the relevant unit is published.

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 AskMQ, 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

1. Weekly tasks in the workshops is a new assessment worth 10%
2. Three module exams instead of six module exams
3. The Major work is replaced by Programming Assessments

Unit information based on version 2023.02 of the Handbook