COMP1000

Introduction to Computer Programming

Session 1, In person-scheduled-weekday, North Ryde 2022

School of Computing

Contents

General Information

Unit convenor and teaching staff
Convenor, Lecturer
Gaurav Gupta
gaurav.gupta@mq.edu.au

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

Content Development Lead
Emilie Jemima Hyslop
emiliejemima.hyslop@mq.edu.au

Tutor
Nataly Falero
nataly.falero@mq.edu.au

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

Tutor
Jessica Dalmati
jessica.dalmati@mq.edu.au

Tutor
Cameron Pappas
cameron.pappas@mq.edu.au

Tutor
Sammy Madafiglio
sammy.madafiglio@mq.edu.au

Tutor
Lucianna Salloum
lucianna.salloum@mq.edu.au

Tutor
Bradley Anderson
bradley.anderson@mq.edu.au

Tutor
Ryan Maehata Fujimoto
ryan.fujimoto@mq.edu.au

Tutor
Alex Taylor
alexander.taylor@mq.edu.au

Tutor
Olivia La Greca
liv.lagreca@mq.edu.au

Tutor
Jack Barnes
jack.barnes@mq.edu.au

Tutor
Hubert Hartan
hubert.hartan@mq.edu.au

Tutor
Sujith Bellam
sujith.bellam@mq.edu.au

Tutor
Rohitranjan Vasantkumar Gupta
rohitranjanvasantkumar.gupta@mq.edu.au

Tutor
Michael Padworth
michael.padworth@mq.edu.au

Tutor
Anthony Ly
anthony.ly@mq.edu.au

Tutor
Izzy Low
izzy.low@mq.edu.au

TUtor
Connor Sanders
connor.sanders@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

Major creative work

Out of the 40 marks for the major creative work,

- 5 marks are for the checkpoint,
- 15 marks are for evaluation of submitted work, and,
- 20 marks are for the following viva.

Module exam hurdle

There are 6 modules, each worth 10% of your total unit mark:

1. Foundations in Programming (FP): Not a hurdle
2. **Variables and Conditions (VC): Hurdle**
3. **Loops (L): Hurdle**
4. **Functions (F): Hurdle**
5. **Compound Data (CD): Hurdle**
6. Program Design and Problem Solving (PDPS): Not a hurdle

You must get 50% or more (5 or more out of 10) in each of modules 2 to 5. You will have two attempts for each module, the first during the semester (see unit schedule) and the second in the final exam period. For each module, the best of the two attempts counts towards your final mark for that module. As an example, if you get 90% in the first attempt for Loops and 40% in the second attempt for Loops, your final mark would be 90% for Loops.

Note that you are NOT required to sit the second attempt if you clear the hurdle for that module in the first attempt. However, you are free to sit the second attempt to try and achieve a higher grade, as we take the higher of the two marks to calculate your final mark for each module.

Late Submission

For the major work, late submissions **will not be accepted** without an approved Special Consideration request. Assessments submitted after the due date will receive a mark of **zero**.

For the module exams, apply for special considerations if you cannot sit the exam during your practical class for any given week.

Assessment Tasks

Name	Weighting	Hurdle	Due
Major Creative Work	40%	No	Checkpoint 5pm 3-Apr. Submission 5pm 29-May, Viva week 13
Module Exams	60%	Yes	Weeks 4, 6, 8, 10, 12, 13 and Final Exam Period

Major Creative Work

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 30 hours
Due: **Checkpoint 5pm 3-Apr. Submission 5pm 29-May, Viva week 13**
Weighting: **40%**

A semester-long programming task where students put all their skills to work creating a game or demo.

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 4, 6, 8, 10, 12, 13 and Final Exam Period**
Weighting: **60%**
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 (even though the latest version is 4.0.5 Beta, we suggest that you install version 3.5.4 to be consistent with the lab. But if you install version 4.0.5, it's also ok!)

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	
4	Variables & Conditionals	variables, debugging in processing, conditions	FP Exams (Week starting 14th March)
5	Loops	loops	
6	Loops		V&C Exams (Week starting 28th March)
7	Functions	functions	Major work Checkpoint (5pm, Sunday 3rd April)
8	Functions	function call dissection, scope	L Exams (Week starting 25th April)
9	Compound Data	compound data	
10	Compound Data	reference semantics	F Exams (Week starting 9th May)
11	Program Design and Problem Solving	refactoring	
12	Program Design and Problem Solving	case study	CD Exams (Week starting 23rd May), Major Creative Work Submission (5pm, Sunday, 29th May)
13	Revision		PDPS exam (Week starting 30th May), Major Creative Work Vivas (Week starting 30th May)
Exam Period			Second-attempts for all six module exams

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 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 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. Nature of hurdles changed from S2, 2021. Modules 2 to 5 are individual hurdles, and there is no collective module exam hurdle requirement.
2. Conditions will initially be introduced in week 3 and expanded upon in week 4.
3. Slight change in distribution of marks for major work.

Changes since First Published

Date	Description
25/03/2022	lecture notes link fixed

Date	Description
19/02/2022	suggested students download processing 3.5.4 to be consistent with labs. add all tutors who got late accounts.