COMP6010

Fundamentals of Computer Science

Session 1, In person-scheduled-weekday, North Ryde 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 (weeks 1 - 6)
Yan Wang
yan.wang@mq.edu.au
Contact via Forum for issues relevant to entire cohort, email for issues specific to you
Please see iLearn

Lecturer (weeks 7 - 12)
Annabelle McIver
annabelle.mciver@mq.edu.au
Contact via Email
Please see iLearn

Credit points
10

Prerequisites

Corequisites

Co-badged status

Unit description
This unit provides a study of algorithms, data structures and programming techniques. The topics covered include: trees; graphs and heaps; advanced sorting techniques; elements of storage management; and complexity. The presentation emphasises the role of data abstraction and correctness proofs.

This unit provides a foundation-level study of programming. The topics covered include programming environment and the process of program execution, variables, operators, boolean logic, control structures including conditions and loops, functions, storage of collection of items and performing operations on the same, and file management.

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 enhanced problem solving skills to develop algorithms

ULO2: Implement programs from algorithms, showing an understanding of control flow.

ULO3: Adhere to standard software development skills such as test-driven development and debugging

ULO4: Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

General Assessment Information

HURDLE

Practical exam 2 is a hurdle task and the best of two attempts (first in week 12 practical class and second in week 13 practical class) will count towards the final grade. You must get at least 50% in practical exam 2 to clear this hurdle. If your total mark in the unit is 50 or more but you do not pass practical exam 2, your final mark will be capped at 49, and your grade will be FH.

Students who pass practical exam 2 in the first attempt (in week 12) cannot take the second attempt.

Students who do not pass practical exam 2 in the first attempt (in week 12) will get a second attempt in week 13 and the mark for practical exam 2 will be capped at 50.

Late Assessment Submission Penalty

Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark) will be applied each day a written 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. Submission time for all written assessments is set at **11:55 pm**. A 1-hour grace period is 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, students need to submit an application for Special Consideration.

Assessments where Late Submissions will be accepted

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

- Weekly Submissions: NO, unless Special Consideration is granted
- Assignments: YES, Standard Late Penalty applies

Assessment Tasks

Name	Weighting	Hurdle	Due
Assignment 1	10%	No	end of Week 4

Name	Weighting	Hurdle	Due
Assignment 2	20%	No	end of Week 13
Practical Exam 1	15%	No	Practical week 8
Practical Exam 2	35%	Yes	Practicals weeks 12, 13
Quizzes	20%	No	Final Exam Period

Assignment 1

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 10 hours
Due: **end of Week 4**
Weighting: **10%**

Programming assignment that requires students to solve a real-life problem based on the contents covered during the semester

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms
- Implement programs from algorithms, showing an understanding of control flow.
- Adhere to standard software development skills such as test-driven development and debugging
- Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

Assignment 2

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **end of Week 13**
Weighting: **20%**

Programming assignment that requires students to solve a real-life problem based on the contents covered during the semester

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms
- Implement programs from algorithms, showing an understanding of control flow.
- Adhere to standard software development skills such as test-driven development and debugging
- Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

Practical Exam 1

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **Practical week 8**
Weighting: **15%**

In-class practical exam assessing contents covered during first half of semester

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms
- Implement programs from algorithms, showing an understanding of control flow.
- Adhere to standard software development skills such as test-driven development and debugging
- Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

Practical Exam 2

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 25 hours
Due: **Practicals weeks 12, 13**
Weighting: **35%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

In-class practical exam assessing contents during the entire semester

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms

- Implement programs from algorithms, showing an understanding of control flow.
- Adhere to standard software development skills such as test-driven development and debugging
- Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

Quizzes

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 20 hours
Due: **Final Exam Period**
Weighting: **20%**

Quizzes over the course of 13 weeks in practical classes (exact weeks to be determined based on timetabling and published in the unit guide).

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms
- Implement programs from algorithms, showing an understanding of control flow.
- Adhere to standard software development skills such as test-driven development and debugging
- Understand and apply important foundation-level programming concepts such as variables, operators, boolean logic, control structures, functions and collections

---

[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

Lecture notes: Check iLearn and also refer to https://softwaretechnologymq.github.io/

Python (3.11.2 at the time of writing this guide): https://python.org/

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

Unit Schedule

- Week 1 - software, installation, the first program

- Week 2 - variables and operators

- Week 3 - conditions

- Week 4 - loops

- Week 5 - problem-solving/case study using topics from weeks 1 to 4

- Week 6 - functions (1)

- Week 7 - problem-solving/case study using topics from weeks 1 to 7

2 WEEK TEACHING BREAK

- Week 8 - collections (1)

- Week 9 - collections (2)

- Week 10 - functions (2) (with collections)

- Week 11 - file system, i/o (raw csv i/o)

- Week 12 - problem-solving/case study using topics from weeks 1 to 11

- Week 13 - no lecture, only practical exam (second attempt, if required)

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 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

None