COMP6010

Fundamentals of Computer Science

Session 1, Online-scheduled-weekday 2022

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
Gaurav Gupta
gaurav.gupta@mq.edu.au
Contact via Forums (or email for issue specific to you)

Lecturer
Yan Wang
yan.wang@mq.edu.au

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

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

Assessment Tasks

Name	Weighting	Hurdle	Due
Weekly submissions	0%	No	Weekly
Online Quiz 1	10%	No	Week 4
Online quiz 2	10%	No	Week 7
Online quiz 3	10%	No	Week 9
Online quiz 4	10%	No	Week 12
Assignment 1	5%	No	5pm, Sunday 27th March
Assignment 2	15%	No	5pm, Sunday 24th April
Assignment 3	10%	No	5pm, Sunday 15th May
Assignment 4	10%	No	5pm, Sunday 5th June
zoom viva	20%	No	Final exam period

Weekly submissions

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 15 hours
Due: **Weekly**
Weighting: **0%**

Each week, students are required to submit a piece of code containing solutions to a given set of problems

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

Online Quiz 1

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 4**
Weighting: **10%**

An online quiz that requires students to complete a number of questions via iLearn within a given time window.

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

Online quiz 2

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 7**
Weighting: **10%**

An online quiz that requires students to complete a number of questions via iLearn within a given time window.

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

Online quiz 3

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 9**
Weighting: **10%**

An online quiz that requires students to complete a number of questions via iLearn within a given time window.

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

Online quiz 4

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 12**
Weighting: **10%**

An online quiz that requires students to complete a number of questions via iLearn within a given time window.

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 1

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 5 hours
Due: **5pm, Sunday 27th March**
Weighting: **5%**

Programming assignment that requires students to solve a real-life problem based on the contents covered in the first half of 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: **5pm, Sunday 24th April**
Weighting: **15%**

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 3

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 10 hours
Due: **5pm, Sunday 15th May**
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 4

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 10 hours
Due: **5pm, Sunday 5th June**
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

zoom viva

Assessment Type [1]: Viva/oral examination
Indicative Time on Task [2]: 12 hours
Due: **Final exam period**
Weighting: **20%**

Students need to answer a number of programming questions, and complete a few programming tasks within a given time window.

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: https://softwaretechnologymq.github.io/

Python (version 3.10.2 at the time of writing this unit guide): https://python.org/

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

Unit Schedule

NOTE: Practical classes begin in week 2. The schedule is for lectures.

- Week 1 - software, installation, variables, operators
- Week 2 - number systems
- Week 3 - boolean logic
- Week 4 - conditions
- Week 5 - loops
- Week 6 - functions 1
- Week 7 - problem-solving/case study using topics from Weeks 1 to 6

TWO-WEEK TEACHING BREAK

- Week 8 - collections* 1
- Week 9 - collections* 2
- Week 10 - classes
- 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 - revision

* one or more from lists, dictionaries, tuples, sets

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

Environment changed from Java to Python

Changes since First Published

Date	Description
04/03/2022	fixed assignment 1 deadline