



# COMP1010

## Fundamentals of Computer Science

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

*School of Computing*

### Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>General Assessment Information</u>	3
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	6
<u>Unit Schedule</u>	7
<u>Policies and Procedures</u>	8

#### 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 and Lecturer (weeks 7-12)

Bernard Mans

[bernard.mans@mq.edu.au](mailto:bernard.mans@mq.edu.au)

TBA

Lecturer (weeks 1-6)

Gaurav Gupta

[gaurav.gupta@mq.edu.au](mailto:gaurav.gupta@mq.edu.au)

TBA

Credit points

10

Prerequisites

(COMP1000 or COMP115) or admission to (BActStud or BActStudBSc or BAppFinBActStud or BActStudBProfPrac)

Corequisites

Co-badged status

Unit description

This unit studies programming as a systematic discipline and introduces more formal software design methods. Programming skills are extended to include elementary data structures and abstract data types. There is a strong emphasis on problem solving and algorithms, including aspects of correctness, complexity and computability.

## 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 the underlying architecture of the computer

**ULO3:** adhere to standard software engineering practices, including documentation, unit testing and debugging

**ULO4:** compare different methods available for the same problem in terms of efficiency and other criteria

**ULO5:** demonstrate foundational learning skills including active engagement in their learning process

## 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
<u>Weekly submissions</u>	20%	Yes	weeks 2-11
<u>Quizzes</u>	20%	No	Weeks 6, 10 and 12 (First 30 minutes of lecture)
<u>End-of-semester exam</u>	30%	No	Weeks 14-15
<u>Assignments</u>	30%	No	Weeks 3, 8 and 13

### Weekly submissions

Assessment Type <sup>1</sup>: Practice-based task

Indicative Time on Task <sup>2</sup>: 20 hours

Due: **weeks 2-11**

Weighting: **20%**

**This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)**

Weekly submissions from the practice package provided. To clear the hurdle, students must submit **at least 8 out of 10** submissions that satisfy the minimum requirements, which may be different for each week.

On successful completion you will be able to:

- apply enhanced problem solving skills to develop algorithms

- implement programs from algorithms, showing an understanding of the underlying architecture of the computer
- compare different methods available for the same problem in terms of efficiency and other criteria
- demonstrate foundational learning skills including active engagement in their learning process

## Quizzes

Assessment Type <sup>1</sup>: Quiz/Test

Indicative Time on Task <sup>2</sup>: 16 hours

Due: **Weeks 6, 10 and 12 (First 30 minutes of lecture)**

Weighting: **20%**

Online quizzes to assess the understanding of the several topics taught throughout the unit. Students get only one attempt for each quiz.

On successful completion you will be able to:

- apply enhanced problem solving skills to develop algorithms
- implement programs from algorithms, showing an understanding of the underlying architecture of the computer
- compare different methods available for the same problem in terms of efficiency and other criteria
- demonstrate foundational learning skills including active engagement in their learning process

## End-of-semester exam

Assessment Type <sup>1</sup>: Examination

Indicative Time on Task <sup>2</sup>: 10 hours

Due: **Weeks 14-15**

Weighting: **30%**

End of semester exam to assess achievement of learning outcomes from the entire session.

On successful completion you will be able to:

- apply enhanced problem solving skills to develop algorithms

- implement programs from algorithms, showing an understanding of the underlying architecture of the computer
- compare different methods available for the same problem in terms of efficiency and other criteria
- demonstrate foundational learning skills including active engagement in their learning process

## Assignments

Assessment Type <sup>1</sup>: Programming Task

Indicative Time on Task <sup>2</sup>: 39 hours

Due: **Weeks 3, 8 and 13**

Weighting: **30%**

Take-home assignments during the semester to assess problem-solving skills in the domain of the unit topics.

On successful completion you will be able to:

- apply enhanced problem solving skills to develop algorithms
- implement programs from algorithms, showing an understanding of the underlying architecture of the computer
- adhere to standard software engineering practices, including documentation, unit testing and debugging
- compare different methods available for the same problem in terms of efficiency and other criteria
- demonstrate foundational learning skills including active engagement in their learning process

---

<sup>1</sup> 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.

<sup>2</sup> 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 hours of lectures (details to be announced via iLearn),
- 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.

Please note that you are **required** to submit work regularly. You will get the help that you need by attending your practical class. Failure to submit work may result in you failing the unit (see the precise requirements in the "Grading Standards" section) or being excluded from the final examination.

### TEXTS AND/OR MATERIALS

Lecture notes, Practical classes and Video tutorials: details to be announced via iLearn

#### Recommended Textbooks:

1. **T. Gaddis, Starting out with Java: From control structures through objects (Pearson), Global edition (6th). ISBN 9781292110653**
  - [Online edition](#) of this book is available through MQ Library. There can be up to 5 simultaneous accesses. Click on "Full text available at: [2018 eTextbooks](#)" and login with OneID and password.
2. **Kathy Siera, Bert Bates, Head First Java, 2nd edition. ISBN 9780596009205**

### TECHNOLOGY USED AND REQUIRED

#### Audio and Video Lecture

Digital recordings of lectures are available from within iLearn via **Active Learning Platform**.

#### Technology

- [Java SE](#) - download the latest Java SE to be compatible with the labs.
- [Eclipse](#) (preferred, troubleshooting provided) or [Visual Studio Code](#) (if you are proficient,

independent) - the IDEs we shall be using during the session.

- Learning Management System [iLearn](#).
- <https://code2flow.com/> for better understanding of control flow.
- <http://codingbat.com/> for programming exercises.

## Discussion Boards

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

## Unit Schedule

Note that three important themes will pervade the entire unit:

1. **Problem-solving.** A crucial skill for all of the weekly topics will be to write appropriate code to meet a given problem specification. This theme relates to the first two learning outcomes for this unit.
2. **Software development.** The use of JUnit testing framework is an important development practice that will be taught from the beginning and used throughout the unit. This theme relates to the third learning outcome of this unit.
3. **Comparing different solution methods.** Very often different algorithms are available for the same problem. Another important skill to develop throughout this unit is the ability to compare different algorithms in terms of efficiency and other criteria. This theme relates to the fourth learning outcome of this unit.

## Unit Schedule

NOTE: This is a tentative schedule and subject to minor changes.

Week	Topic	Pre-lecture readings from <a href="#">Lecture Notes</a> (COMP1010)	Assessments Due	Weekly submission
1	Programming environment	1, 2		No
2	Problem-solving, JUnit testing	3, 4		Yes
3	Classes and Objects - 1	5, 6, 7	Assignment 1 (Arrays) - 5%	Yes
4	Classes and Objects - 2	8, 9, 10		Yes

5	Recursion - 1	11, 12		Yes
6	Recursion - 2	13, 14, 15	Quiz 1 (Classes) - 5% First 30 minutes of lecture	Yes
7	List Interface, ArrayList class	16, 17, 18		Yes
<b>2 weeks' recess</b>				
8	Sorting Algorithm(s)	TO DO	Assignment 2 (Classes) - 10%	Yes
9	Stacks and Queues	TO DO		Yes
10	Recursive data structures - 1	21	Quiz 2 (Recursion) – 7% First 30 minutes of lecture	Yes
11	Recursive data structures - 2	22		Yes
12	Assembly Programming	TBA	Quiz 3 (ArrayList class) - 8% First 30 minutes of lecture	No
13	-		Assignment 3 (Recursive data structures) - 15%	No
14, 15			Final Exam – 30%	No

## Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central \(https://policies.mq.edu.au\)](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 \(https://students.mq.edu.au/su\)](https://students.mq.edu.au/su)



[upport/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](mailto:ask.mq.edu.au) or if you are a Global MBA student contact [globalmba.support@mq.edu.au](mailto: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/](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.