



COMP6010

Fundamentals of Computer Science

Session 2, Special circumstance 2020

Department 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> | 8 |
| <u>Unit Schedule</u> | 9 |
| <u>Policies and Procedures</u> | 10 |
| <u>Changes from Previous Offering</u> | 12 |
| <u>Grading Standards</u> | 12 |

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.

Notice

As part of [Phase 3 of our return to campus plan](#), most units will now run tutorials, seminars and other small group learning activities on campus for the second half-year, while keeping an online version available for those students unable to return or those who choose to continue their studies online.

To check the availability of face-to-face and online activities for your unit, please go to [timetable viewer](#). To check detailed information on unit assessments visit your unit's iLearn space or consult your unit convenor.

General Information

Unit convenor and teaching staff

Convenor, Lecturer

Yipeng Zhou

yipeng.zhou@mq.edu.au

Contact via yipeng.zhou@mq.edu.au

Room 371 4 Research Park Drive, Macquarie Park

Tue. 10:00-12:00

Lecturer

Gaurav Gupta

gaurav.gupta@mq.edu.au

Contact via gaurav.gupta@mq.edu.au

To be advised

Tutor

Aliyah Ali

aliyah.ali@students.mq.edu.au

Contact via Email

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.

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

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

General Assessment Information

Late Submission

No extensions will be granted without an approved application for Special Consideration. There will be a deduction of 20% of the total available marks made from the total awarded mark for each 24 hour period or part thereof that the submission is late. For example, 25 hours late in submission for an assignment worth 10 marks – 40% penalty or 4 marks deducted from the total. No submission will be accepted after solutions have been posted.

Assessment Tasks

| Name | Weighting | Hurdle | Due |
|-------------------------------|-----------|--------|---------|
| Assignment 1 | 5% | No | Week 4 |
| Assignment 2 | 15% | No | Week 6 |
| Assignment 3 | 10% | No | Week 11 |
| Assignment 4 | 10% | No | Week 13 |
| Online Quiz 1 | 10% | No | Week 5 |
| Online quiz 2 | 10% | No | Week 6 |
| Online quiz 3 | 10% | No | Week 12 |
| Online quiz 4 | 10% | No | Week 13 |
| zoom viva | 20% | No | Week 14 |

Assignment 1

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 5 hours

Due: **Week 4**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Assignment 2

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 15 hours

Due: **Week 6**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Assignment 3

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 10 hours

Due: **Week 11**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Assignment 4

Assessment Type ¹: Programming Task

Indicative Time on Task ²: 10 hours

Due: **Week 13**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Online Quiz 1

Assessment Type ¹: Quiz/Test

Indicative Time on Task ²: 5 hours

Due: **Week 5**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Online quiz 2

Assessment Type ¹: Quiz/Test

Indicative Time on Task ²: 5 hours

Due: **Week 6**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Online quiz 3

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

Online quiz 4

Assessment Type ¹: Quiz/Test

Indicative Time on Task ²: 5 hours

Due: **Week 13**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

zoom viva

Assessment Type ¹: Viva/oral examination

Indicative Time on Task ²: 12 hours

Due: **Week 14**

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 the underlying architecture of the computer
- Adhere to standard software engineering practices
- Compare different methods available for the same problem in terms of efficiency and other criteria

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

The unit will be delivered online. Each week you should attend

- two hours of online lectures, and,
- two hours of online workshops, and,
- two hour online practical classes

For details of days, times and rooms consult the [timetables webpage](#).

Note that Lectures, Workshops and Practical Classes commence in week 1.

You should have selected a practical class during enrolment. **You should attend the practical class you are enrolled in.** If you do not have a class, or if you wish to change one, you should see the enrolment operators in the E7B courtyard during the first two weeks of the semester. Thereafter you should go to the Science and Engineering Student Services Centre.

Please note that you are **required** to submit work and take online quiz regularly. You will get the help that you need by attending your workshop. 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.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

Textbook

The first book in the following list, namely ***Starting out with Java***, is the recommended text book. The other books listed are helpful references.

- T. Gaddis, [Starting out with Java: From control structures through objects](#) (Pearson), Global edition (6th). ISBN 9781292110653
- W. Savitch, [Absolute Java](#) (Pearson) 5th edition. ISBN 9780132830317
- B. Eckel, [Thinking in Java](#) (electronic book, 3rd edition available within iLearn is fine and is free but does not cover data structures)
- A. Drozdek, [Data Structures and Algorithms in Java](#) (Cengage) 2nd edition. ISBN 9780534492526 (this book will also be used in COMP225)
- D. Carlson, [Eclipse Distilled](#) (Addison-Wesley) 1st edition. ISBN 9780321288158 (extensive coverage of the software development platform eclipse)

TECHNOLOGY USED AND REQUIRED

Online Delivery

All online lectures, workshops and practical classes will be delivered via zoom. Meanwhile, all zoom sessions will be recorded as audio lecture.

Audio Lecture

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

Technology

- [Eclipse](#) - download Eclipse IDE for Java Developers
- [Java SE](#) - download Java SE 9 (8 is also fine) to be compatible with the labs.
- Learning Management System [iLearn](#)
- <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. Use of the JUnit testing framework is an important development practice which 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.

| WEEK | TOPIC |
|------|----------------------------|
| 1 | Introduction, fundamentals |
| 2 | Basics of programming |
| 3 | Control Structures |
| 4 | Functions - 1 |
| 5 | Arrays - 1 |
| 6 | Functions and arrays |
| 7 | Case study |
| 8 | Classes and objects |
| 9 | Lists and Maps |
| 10 | Time complexity |
| 11 | Useful algorithms |
| 12 | Low-level programming |
| 13 | Review |

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central \(https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central\)](https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central). 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](#)
- [Grade Appeal Policy](#)
- [Complaint Management Procedure for Students and Members of the Public](#)
- [Special Consideration Policy](#) (**Note:** *The Special Consideration Policy is effective from 4 December 2017 and replaces the Disruption to Studies Policy.*)

Students seeking more policy resources can visit the [Student Policy Gateway](https://students.mq.edu.au/support/study/student-policy-gateway) (<https://students.mq.edu.au/support/study/student-policy-gateway>). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

If you would like to see all the policies relevant to Learning and Teaching visit [Policy Central](http://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central) (<http://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central>).

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: <https://students.mq.edu.au/study/getting-started/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

Student Support

Macquarie University provides a range of support services for students. For details, visit <http://students.mq.edu.au/support/>

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to help you improve your marks and take control of your study.

- [Getting help with your assignment](#)
- [Workshops](#)
- [StudyWise](#)
- [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

Students with a disability are encouraged to contact the [Disability Service](#) who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

If you are a Global MBA student contact globalmba.support@mq.edu.au

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

All lectures, workshops and practical classes are delivered online via zoom..

Practical exams are removed from assessment tasks. Online quiz, more assignments and zoom viva are added as assessment tasks.

Grading Standards

Three standards, namely Developing, Functional, and Proficient, summarize as many different levels of achievement. Each standard is precisely defined to help students know what kind of performance is expected to deserve a certain mark. The standards corresponding to the learning outcomes of this unit are given below:

| | | | |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L.O. #1 | Limited ability to solve problems. Limited knowledge of basic data structures. | Ability to write simple algorithms and solve simple problems using OOD. Know basic data structures (queues, stacks, linked lists) and how to manipulate them. | Ability to write complex algorithms and solve complex problems using OOD and recursion. Ability to select the most appropriate data structures to solve a problem. |
| L.O. #2 | Show poor programming skills. Limited ability to write code that compiles or executes properly. | Show basic programming skills. Understand notions of compiler and virtual machine. Know types, how to implement simple conditions, simple loops, simple data structures, simple objects. | Show advanced programming skills. Understand notions of compiler and virtual machine. Know types, how to implement conditions, loops, data structures, objects. Understand inheritance and polymorphism. |
| L.O. #3 | Unability to follow specifications. Poor coding style. Poor documentation. Submission of incorrect programs showing no sign of testing/debugging skills. | Follow simple specifications. Document code (e.g. pre-post conditions). Test and debug a simple program. Understand the notion of modularity/object file. | Understand the importance of specifications. Neat code/consistent programming style. Clear and insightful comments. Design test cases and debug programs. |

At the end of the semester, you will receive a grade that reflects your achievement in the unit.

- **Fail (F):** does not provide evidence of attainment of all learning outcomes. There is missing or partial or superficial or faulty understanding and application of the fundamental concepts in the field of study; and incomplete, confusing or lacking communication of ideas in ways that give little attention to the conventions of the discipline.
- **Pass (P):** provides sufficient evidence of the achievement of learning outcomes. There is demonstration of understanding and application of fundamental concepts of the field of study; and communication of information and ideas adequately in terms of the conventions of the discipline. The learning attainment is considered satisfactory or adequate or competent or capable in relation to the specified outcomes.
- **Credit (Cr):** provides evidence of learning that goes beyond replication of content knowledge or skills relevant to the learning outcomes. There is demonstration of substantial understanding of fundamental concepts in the field of study and the ability to apply these concepts in a variety of contexts; plus communication of ideas fluently and clearly in terms of the conventions of the discipline.
- **Distinction (D):** provides evidence of integration and evaluation of critical ideas, principles and theories, distinctive insight and ability in applying relevant skills and concepts in relation to learning outcomes. There is demonstration of frequent originality in defining and analysing issues or problems and providing solutions; and the use of means of communication appropriate to the discipline and the audience.
- **High Distinction (HD):** provides consistent evidence of deep and critical understanding in relation to the learning outcomes. There is substantial originality and insight in identifying, generating and communicating competing arguments, perspectives or problem solving approaches; critical evaluation of problems, their solutions and their implications; creativity in application.

In this unit, your final grade depends on your performance for each component of the assessment. Indeed, for each task, you receive a mark that captures your standard of performance regarding each learning outcome assessed by this task. Then the different component marks are added up to determine your total mark out of 100. Your grade then depends on this total mark and your overall standard of performance.

Concretely, **you will pass the unit**, if

- your total mark is at least 50 out of 100

In order to obtain a higher grade than a Pass, you must fulfil the conditions for a Pass and satisfy the following requirements,

- at least 85% overall for High Distinction

- at least 75% overall for Distinction;
- at least 65% overall or Credit.