COMP6010

Fundamentals of Computer Science

Session 2, Fully online/virtual 2021

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.

Session 2 Learning and Teaching Update

The decision has been made to conduct study online for the remainder of Session 2 for all units WITHOUT mandatory on-campus learning activities. Exams for Session 2 will also be online where possible to do so.

This is due to the extension of the lockdown orders and to provide certainty around arrangements for the remainder of Session 2. We hope to return to campus beyond Session 2 as soon as it is safe and appropriate to do so.

Some classes/teaching activities cannot be moved online and must be taught on campus. You should already know if you are in one of these classes/teaching activities and your unit convenor will provide you with more information via iLearn. If you want to confirm, see the list of units with mandatory on-campus classes/teaching activities.

Visit the MQ COVID-19 information page for more detail.

General Information

Unit convenor and teaching staff
Yipeng Zhou
yipeng.zhou@mq.edu.au

Gaurav Gupta
gaurav.gupta@mq.edu.au

Tutor
Naila Mukhtar
naila.mukhtar@mq.edu.au
Contact via email

Credit points
10

Prerequisites

Corequisites

Co-badged status

Unit description
*This unit has an online offering for S2 which is **synchronous**, meaning there will be set times to attend online lectures and tutorials.*

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

Assessment Tasks

Name	Weighting	Hurdle	Due
Assignment 1	5%	No	Week 4
Assignment 2	15%	No	Week 6
Assignment 3	10%	No	Week 9
Assignment 4	10%	No	Week 13
Online Quiz 1	10%	No	Week 5
Online quiz 2	10%	No	Week 7
Online quiz 3	10%	No	Week12
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 9**
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 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 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: **Week12**
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. 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.

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
- Grade Appeal Policy
- Complaint Management 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

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.