COMP1010

Fundamentals of Computer Science

Session 2, Special circumstance 2020

Department of Computing

Contents

General Information 2

Learning Outcomes 3

General Assessment Information 3

Assessment Tasks 4

Delivery and Resources 9

Unit Schedule 10

Policies and Procedures 11

Changes from Previous Offering 13

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 ot her small group learning activities on campus for the second half-year, while keeping an online ver sion available for those students unable to return or those who choose to continue their studies onli ne.

To check the availability of face-to-face and onlin e activities for your unit, please go to timetable vi ewer. To check detailed information on unit asses sments visit your unit's iLearn space or consult yo ur unit convenor.

General Information

Unit convenor and teaching staff
Convenor, Lecturer
Gaurav Gupta
gaurav.gupta@mq.edu.au
Contact via Contact via Dialogue Utility on iLearn
To be advised via iLearn

Tutor
Daniel Sutantyo
daniel.sutantyo@mq.edu.au

Tutor
John Kim
j.kim@mq.edu.au

Tutor
Michael Lay
michael.lay@mq.edu.au

Tutor
Sophie Kaelin
sophie.kaelin@mq.edu.au

Tutor
Bradley Kenny
bradley.kenny@mq.edu.au

Tutor
Samantha Kuhn
samantha.kuhn@mq.edu.au

Tutor
Cooper Timewell
cooper.timewell@mq.edu.au

Support tutor
Cecilia Hollins
cecilia.hollins@mq.edu.au

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 Submission

No extensions will be granted without an approved application for Special Consideration. There will be a deduction of 20% of the assignment 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 the solutions have been posted.

Assessment Tasks

Name	Weighting	Hurdle	Due
Online quiz 1	10%	No	Week 7
Online quiz 2	10%	No	Week 9
Online quiz 3	10%	No	Week 11
Online quiz 4	10%	No	Week 13
Assignment 1	10%	No	Week 3
Assignment 2	10%	No	Week 6
Assignment 3	10%	No	Week 10
Assignment 4	10%	No	Week 13
End-of-semester exam	20%	No	Exam period

Online quiz 1

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

Online quiz to assess knowledge of classes and objects

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

Online quiz 2

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

Quiz to assess understanding of recursion.

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

Online quiz 3

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 11**
Weighting: **10%**

Quiz to assessment understanding of the built-in ArrayList class

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

Online quiz 4

Assessment Type [1]: Quiz/Test
Indicative Time on Task [2]: 5 hours
Due: **Week 13**
Weighting: **10%**

Quiz to assess understanding of custom-built data structures and recursive data structures

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

Assignment 1

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 8 hours
Due: **Week 3**
Weighting: **10%**

Assignment to assess problem solving skills, understanding of primitive data types and operations, control structures, arrays and methods.

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

Assignment 2

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 8 hours
Due: **Week 6**
Weighting: **10%**

Assignment to assess problem solving skills and understanding of classes and objects.

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

Assignment 3

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 8 hours
Due: **Week 10**
Weighting: **10%**

Assignment to assess problem-solving using built-in ArrayList class.

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

Assignment 4

Assessment Type [1]: Programming Task
Indicative Time on Task [2]: 8 hours
Due: **Week 13**
Weighting: **10%**

Assignment to assess problem solving skills using recursive data structures.

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

End-of-semester exam

Assessment Type [1]: Examination
Indicative Time on Task [2]: 14 hours
Due: **Exam period**
Weighting: **20%**

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
- 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

---

[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

CLASSES

Each week you should attend

- three hours of lectures (delivered online, 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: https://rebrand.ly/COMP1010LectureNotes

Video tutorials: https://rebrand.ly/COMP1010VideoTutorials

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 and Visual Studio Code - the IDEs we shall be using during the session.
- 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.

Week	Topic	Assessments
1	Programming environment (language, IDEs)	Diagnostic test (does not contribute towards final mark)
2	Problem-solving, JUnit testing	
3	Classes and Objects - 1	Assignment 1 Due
4	Classes and Objects - 2	
5	Recursion - 1	
6	Binary search, Merge Sort	Assignment 2 Due
7	List Interface, ArrayList class	Online Quiz 1
8	Iterators	
9	Custom-built ArrayList	Online Quiz 2
10	Self-referencing classes (Node class)	Assignment 3 Due
11	Custom-built LinkedList	Online Quiz 3
12	Advanced topics	
13	Revision	Online quiz 4, Assignment 4

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). 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). 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 (https://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

Due to COVID-restrictions, the assessment structure has significantly changed. Please see the section on Assessments Tasks for further details.

There are no hurdle assessments.