COMP6010
Fundamentals of Computer Science
Session 1, Weekday attendance, North Ryde 2020
Dept of Computing

Contents

General Information 2
Learning Outcomes 2
General Assessment Information 3
Assessment Tasks 3
Delivery and Resources 6
Unit Schedule 7
Policies and Procedures 8
Changes from Previous Offering 10

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
Gaurav Gupta
gaurav.gupta@mq.edu.au

Yipeng Zhou
yipeng.zhou@mq.edu.au

Tutor
Shahpar Yakhchi
shahpar.yakhchi@mq.edu.au

Tutor
Seyed Ghafari
seyed-mohssen.ghafari@mq.edu.au

Content Developer
Michael Lay
michael.lay@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.

Important Academic Dates
Information about important academic dates including deadlines for withdrawing from units are available at https://students.mq.edu.au/important-dates

Learning Outcomes

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

Students with reasonable adjustments

Students with reasonable adjustment approvals will sit practical exams in relevant weeks on Friday 18:00.

Practical Exam 2

For practical exam 2, the best (out of 2 attempts) mark counts towards the final grade.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Weighting</th>
<th>Hurdle</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly submissions</td>
<td>0%</td>
<td>No</td>
<td>Weekly</td>
</tr>
<tr>
<td>Practical Exam 1</td>
<td>15%</td>
<td>No</td>
<td>Week 7</td>
</tr>
<tr>
<td>Practical Exam 2</td>
<td>25%</td>
<td>Yes</td>
<td>Week 13, 14</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>5%</td>
<td>No</td>
<td>Week 6</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>15%</td>
<td>No</td>
<td>Week 12</td>
</tr>
<tr>
<td>Final Examination</td>
<td>40%</td>
<td>No</td>
<td>Exam Period</td>
</tr>
</tbody>
</table>

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
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

Practical Exam 1
Assessment Type: Programming Task
Indicative Time on Task: 10 hours
Due: Week 7
Weighting: 15%

In-class practical exam assessing contents covered during first half of 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

Practical Exam 2
Assessment Type: Programming Task
Indicative Time on Task: 15 hours
Due: Week 13, 14
Weighting: 25%
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

In-class practical exam assessing contents covered during the entire 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 1
Assessment Type 1: Programming Task
Indicative Time on Task 2: 5 hours
Due: Week 6
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 12
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

Final Examination
Assessment Type 1: Examination
Indicative Time on Task: 12 hours
Due: Exam Period
Weighting: 40%
2-hour written exam

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 guidance or support to understand or complete this type of assessment, please contact the Learning Skills Team

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

- two hours of lectures, and,
- two hour practicals,

Additionally, a two-hour workshop is held that serves as consultation hours.

For details of days, times and rooms consult the timetables webpage.

Note that Lectures and Workshops commence in week 1.

You should have selected a workshop during enrolment. You should attend the workshop 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 14 Sir Christopher Ondaatje avenue 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 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 textbook. The other books listed are helpful references.

- **B. Eckel**, *Thinking in Java* (electronic book, 3rd edition available within iLearn is fine and is free but does not cover data structures)

TECHNOLOGY USED AND REQUIRED

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

Technology

- **Java SE** - download Java SE 9 (8 is also fine) to be compatible with the labs.
- **Eclipse** and **Visual Studio Code** - IDEs used.
- 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.

Tentative teaching schedule, subject to change:

<table>
<thead>
<tr>
<th>WEEK</th>
<th>TOPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, fundamentals</td>
</tr>
<tr>
<td>2</td>
<td>Basics of programming</td>
</tr>
<tr>
<td>3</td>
<td>Control Structures</td>
</tr>
<tr>
<td>4</td>
<td>Functions - 1</td>
</tr>
<tr>
<td>5</td>
<td>Arrays - 1</td>
</tr>
<tr>
<td>6</td>
<td>Functions and arrays</td>
</tr>
<tr>
<td>7</td>
<td>Case study</td>
</tr>
<tr>
<td>8</td>
<td>Classes and objects</td>
</tr>
<tr>
<td>9</td>
<td>Lists and Maps</td>
</tr>
<tr>
<td>10</td>
<td>Time complexity</td>
</tr>
<tr>
<td>11</td>
<td>Useful algorithms</td>
</tr>
<tr>
<td>12</td>
<td>Computer Organization</td>
</tr>
<tr>
<td>13</td>
<td>Review</td>
</tr>
</tbody>
</table>

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 improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

Student Enquiry Service

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

Equity 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.

IT Help

For help with University computer systems and technology, visit http://www.mq.edu.au/about_us/
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**

1. week 12 topic changed from "low level programming" to "computer organization".  
2. two attempts offered for the hurdle Practical Exam 2 to all students.