



# COMP125

## Fundamentals of Computer Science

S2 Day 2014

*Computing*

### Contents

---

<a href="#"><u>General Information</u></a>	2
<a href="#"><u>Learning Outcomes</u></a>	3
<a href="#"><u>General Assessment Information</u></a>	3
<a href="#"><u>Assessment Tasks</u></a>	3
<a href="#"><u>Delivery and Resources</u></a>	6
<a href="#"><u>Unit Schedule</u></a>	8
<a href="#"><u>Policies and Procedures</u></a>	8
<a href="#"><u>Graduate Capabilities</u></a>	10
<a href="#"><u>Changes from Previous Offering</u></a>	11
<a href="#"><u>Grading Standards</u></a>	11

---

#### **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

Convener, Lecturer

Steve Cassidy

[steve.cassidy@mq.edu.au](mailto:steve.cassidy@mq.edu.au)

Contact via 98509581

E7A377

By Appointment

Lecturer

Scott McCallum

[scott.mccallum@mq.edu.au](mailto:scott.mccallum@mq.edu.au)

Contact via 98509575

E6A375

Tutor

Gaurav Gupta

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

Tutor

Matt Cabanag

[matt.cabanag@mq.edu.au](mailto:matt.cabanag@mq.edu.au)

Tutor

Oldooz Dianat

[Oldooz.Dianat@mq.edu.au](mailto:Oldooz.Dianat@mq.edu.au)

Tutor

Kym Haines

[kym.haines@mq.edu.au](mailto:kym.haines@mq.edu.au)

Tutor

Luke Mathieson

[luke.mathieson@mq.edu.au](mailto:luke.mathieson@mq.edu.au)

Tutor

Amara Atif

[Amara.Atif@mq.edu.au](mailto:Amara.Atif@mq.edu.au)

Credit points

3

#### Prerequisites

COMP115(P) or COMP155(P) or [admission to (BCom-ActStud or BActStud or BAdvSc or Advanced Program BSc) or (an equivalent admission rank or aggregate)]

#### 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:

Apply enhanced problem solving skills to develop algorithms

Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer

Follow standard software engineering practices (in particular document, test and debug programs)

## General Assessment Information

### Late Submission

Late submission of assignments will be accepted but will attract a penalty of 10% for each day (24hrs) they are late. So, if you submit 25hrs after the deadline, you will only be able to get 80% of the available marks for the task.

If you are submitting late because of an unavoidable disruption, please submit a [Disruption to Studies](#) request with appropriate evidence.

## Assessment Tasks

Name	Weighting	Due
<a href="#">Weekly exercise</a>	10%	End of every week

Name	Weighting	Due
<a href="#">Diagnostic Quiz</a>	2%	Week 4
<a href="#">Assignment 1</a>	6%	Week 5
<a href="#">Assignment 2</a>	10%	Week 8
<a href="#">Assignment 3</a>	12%	Week 12
<a href="#">Mid Semester Test</a>	10%	Wed 17 September
<a href="#">Final examination</a>	50%	TBA

## Weekly exercise

Due: **End of every week**

Weighting: **10%**

Each week, an online quiz or a programming exercise will be made available online. Each task is worth 1 mark. A maximum of 10 marks will count towards the assessment of the unit.

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
- Follow standard software engineering practices (in particular document, test and debug programs)

## Diagnostic Quiz

Due: **Week 4**

Weighting: **2%**

You must complete a randomised online quiz available at <https://iq.science.mq.edu.au/> by the end of week 4.

You may attempt this task as many times as necessary until you pass it. Students who do not pass this task will not pass the unit. To pass the quiz you must get **all questions correct**.

On successful completion you will be able to:

- Apply enhanced problem solving skills to develop algorithms

## Assignment 1

Due: **Week 5**

Weighting: **6%**

A programming assignment that will require you to write Java code to meet a set of requirements. Your code will be assessed via automated tests that will be provided to you. You will also be marked on code quality and completeness.

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
- Follow standard software engineering practices (in particular document, test and debug programs)

## Assignment 2

Due: **Week 8**

Weighting: **10%**

A more involved programming assignment that will involve the use of arrays and algorithms for searching and/or sorting. Your code will be assessed via automated tests that will be provided to you. You will also be marked on code quality and completeness.

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
- Follow standard software engineering practices (in particular document, test and debug programs)

## Assignment 3

Due: **Week 12**

Weighting: **12%**

An advanced programming assignment where you will be asked to build a more complex application that will involve the use of some of the data structures and algorithms discussed in the unit. Your code will be assessed via automated tests that will be provided to you. You will also be marked on code quality and completeness.

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

- Follow standard software engineering practices (in particular document, test and debug programs)

## Mid Semester Test

Due: **Wed 17 September**

Weighting: **10%**

A 50-minute written examination held during class time will test your understanding of material covered in weeks 1 to 6. You will receive some feedback that will allow you to be better prepared for the final examination.

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

## Final examination

Due: **TBA**

Weighting: **50%**

A three hour written examination.

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

## Delivery and Resources

### CLASSES

Each week you should attend

- three hours of lectures and
- a two-hour workshop (or mixed class), i.e. a tutorial and a practical combined in a single session.

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

**Note that 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 E7B courtyard during the first two weeks of the semester. Thereafter you should go to the Student Centre.

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

## HELP101

A help desk where tutors are available for consultation on 100-level units.

## REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

### Textbook

There isn't any prescribed textbook for COMP125 this semester. We suggest below a few references if you feel you need to complement the lecture notes that are provided:

- W. Savitch, [Absolute Java](#) (Pearson) 5th edition. ISBN 9780132830317 (covers basic Java programming and data structures, ideal for COMP125 but not beyond. We will follow this book as much as we can.)
- 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

### Audio Lecture

Digital recordings of lectures are available from within iLearn via [Echo360](#).

### Technology

- [Eclipse](#) - download Eclipse IDE for Java Developers
- [Java SE JDK](#) - download Java SE 7 to be compatible with the labs
- Note that you need the Java JDK 7 which includes the compiler tools. Make sure that you also the Java Runtime Environment JRE 7 to allow you to run Java applications.
- Learning Management System [iLearn](#)
- [iQ system](#) for the diagnostic task

### Discussion Boards

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

## Unit Schedule

1	Introducing Java	Scott
2	Developing and Testing Java programs	Scott
3	Arrays and Simple Searching	Scott
4	Classes and Objects	Scott
5	Arrays of Objects, Search Algorithms	Scott
6	Sorting Algorithms	Scott
7	Software Development + Mid Semester Test	Steve
	Mid Semester Break	
8	Container Types	Steve
9	Recursive Algorithms	Steve
10	Recursive Data Structures	Steve
11	Stack and Queues	Steve
12	Object Oriented Design & Development	Steve
13	Review	Steve + Scott

## Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central](#). Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy [http://mq.edu.au/policy/docs/academic\\_honesty/policy.html](http://mq.edu.au/policy/docs/academic_honesty/policy.html)

Assessment Policy <http://mq.edu.au/policy/docs/assessment/policy.html>

Grading Policy <http://mq.edu.au/policy/docs/grading/policy.html>

Grade Appeal Policy <http://mq.edu.au/policy/docs/gradeappeal/policy.html>

Grievance Management Policy [http://mq.edu.au/policy/docs/grievance\\_management/policy.html](http://mq.edu.au/policy/docs/grievance_management/policy.html)

Disruption to Studies Policy [http://www.mq.edu.au/policy/docs/disruption\\_studies/policy.html](http://www.mq.edu.au/policy/docs/disruption_studies/policy.html) *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

In addition, a number of other policies can be found in the [Learning and Teaching Category](#) of



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/support/student\\_conduct/](https://students.mq.edu.au/support/student_conduct/)

Department of Computing Special Consideration Policy [http://comp.mq.edu.au/undergrad/policies/special\\_consideration\\_policy.htm](http://comp.mq.edu.au/undergrad/policies/special_consideration_policy.htm)

### Special Consideration

[Special Consideration](#) is intended for a student who is prevented by serious and unavoidable disruption from completing any unit requirements in accordance with their ability. To apply for special consideration you need to use the [online submission system](#) along with some evidence to support your case. Depending on the circumstances presented, the convenor may choose to give you an alternate assessment, additional time for an assessment, make-up exam, etc. If a Supplementary Examination is granted as a result of the Special Consideration process the examination will be scheduled after the conclusion of the official examination period. For details of the Special Consideration policy specific to the Department of Computing, see the Department's [policy page](#).

## 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](http://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 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](http://ask.mq.edu.au)

## IT Help

For help with University computer systems and technology, visit <http://informatics.mq.edu.au/help/>.

When using the University's IT, you must adhere to the [Acceptable Use Policy](#). The policy

applies to all who connect to the MQ network including students.

## **Graduate Capabilities**

### **Discipline Specific Knowledge and Skills**

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

#### **Learning outcomes**

- Apply enhanced problem solving skills to develop algorithms
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Follow standard software engineering practices (in particular document, test and debug programs)

#### **Assessment tasks**

- Weekly exercise
- Diagnostic Quiz
- Assignment 1
- Assignment 2
- Assignment 3
- Mid Semester Test
- Final examination

### **Problem Solving and Research Capability**

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

#### **Learning outcomes**

- Apply enhanced problem solving skills to develop algorithms

- Follow standard software engineering practices (in particular document, test and debug programs)

## **Assessment tasks**

- Weekly exercise
- Diagnostic Quiz
- Assignment 1
- Assignment 2
- Assignment 3
- Mid Semester Test
- Final examination

## **Creative and Innovative**

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

## **Assessment tasks**

- Weekly exercise
- Assignment 1
- Assignment 2
- Assignment 3

## **Changes from Previous Offering**

Based on feedback from last year's offerings we will be paying more attention to the transition from COMP115 and Processing to the use of Java and Eclipse in COMP125. This will include some specific exercises in the first weeks of class and the use of the Processing libraries when writing Java applications. Peer tutors are students who have recently completed COMP125 who will be present in Workshops for part of the semester to give help and support to students. They bring their recent experience in the unit and help to ensure that more students get the attention they need during the workshop.

## **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; and
- you pass the mid term test + final examination (i.e. obtain an aggregate mark of at least 30 out of 60); and
- you pass the diagnostic quiz.

In order to obtain a higher grade than a Pass, you must fulfill the conditions for a Pass and have a total mark of at least

- 85 for High Distinction;
- 75 for Distinction;
- 65 for Credit.