COMP125

Fundamentals of Computer Science

S2 Day 2013

Computing

Contents

General Information 2

Learning Outcomes 4

Assessment Tasks 4

Delivery and Resources 8

Unit Schedule 9

Policies and Procedures 10

Graduate Capabilities 11

Grading Standards 13

What has changed since last offering 15

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
Tutor
Amara Atif
Amara.Atif@mq.edu.au
Contact via Amara.Atif@mq.edu.au

Tutor
Matt Cabanag
matt.cabanag@mq.edu.au
Contact via matt.cabanag@mq.edu.au
E6A 369
By appointment

Unit Convenor
Christophe Doche
christophe.doche@mq.edu.au
Contact via christophe.doche@mq.edu.au

Lecturer
Scott McCallum
scott.mccallum@mq.edu.au
Contact via scott.mccallum@mq.edu.au
E6A 375
By appointment

Tutor
Malin Md Mokammel Haque
malin.mokammel-haque@mq.edu.au
Contact via malin.mokammel-haque@mq.edu.au

Tutor
Md Abeed Sarker
abeed.sarker@mq.edu.au
Contact via abeed.sarker@mq.edu.au

Tutor
Scott Buckley
scott.buckley@mq.edu.au
Contact via scott.buckley@mq.edu.au

Tutor
Pongsak Suvanpong
pongsak.suvanpong@mq.edu.au
Contact via pongsak.suvanpong@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:

Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills

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

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

Assessment Tasks

Name	Weighting	Due
Weekly exercise	10%	End of every week
Workshop participation	3%	Ongoing
Diagnostic Quiz	2%	Week 4
Assignment 1	5%	Week 4

Name	Weighting	Due
Assignment 2	10%	Week 8
Assignment 3	10%	Week 12
Mid-term test	10%	Week 7
Final examination	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:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

Workshop participation

Due: **Ongoing**
Weighting: **3%**

Attendance and active participation to the workshop contribute to your final grade.

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

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 are not able to pass this task are not allowed to sit the final examination.

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer

Assignment 1

Due: **Week 4**
Weighting: **5%**

There are three assignments that are worth 25% in total. Each assignment includes different components to assess different levels of achievements.

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

Assignment 2

Due: **Week 8**
Weighting: **10%**

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying

architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

Assignment 3

Due: **Week 12**
Weighting: **10%**

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

Mid-term test

Due: **Week 7**
Weighting: **10%**

A 50-minute written examination worth 10% will be held in week 7 during class time. It 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:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

Final examination

Due: **TBA**
Weighting: **50%**

A three hour examination worth 50% will be held at the end of the semester. More details on the

structure of the examination will be provided towards the end of the semester.

On successful completion you will be able to:

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying architecture of the computer
- Implement programs following standard software engineering practices (in particular document, test and debug)

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)

UNIT WEBPAGE AND 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

See the resources page for more details.

Websites

The web page for this unit can be found at http://www.comp.mq.edu.au/units/comp125

Discussion Boards

The unit makes use of forums hosted within iLearn. Please post questions there, they are monitored by the staffs of the unit.

Unit Schedule

1	Introducing Java	Scott
2	Developing and Testing Java programs	Scott
3	Object Oriented Design and Development	Scott
4	Search Algorithms	Scott
5	Sorting Algorithms	Scott

6	Software Development Topics	Scott
7	Mid-term Test and Container Types	Christophe
8	Introduction to Recursion	Christophe
9	More on Recursion	Christophe
10	Linked Lists	Christophe
11	Stack and Queues	Christophe
12	Inheritance and Polymorphism	Christophe
13	Review	Both

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://www.mq.edu.au/policy/docs/academic_honesty/policy.html

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

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

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

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Special Consideration Policy http://www.mq.edu.au/policy/docs/special_consideration/policy.html

In addition, a number of other policies can be found in the Learning and Teaching Category of Policy Central.

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 submit an application using ask 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.

Late assignment

No extensions will be granted. Students who have not submitted the task prior to the deadline will be awarded a mark of 0 for the task, except for cases in which an application for special consideration is made and approved.

Student Support

Macquarie University provides a range of Academic Student Support Services. Details of these services can be accessed at: http://students.mq.edu.au/support/

UniWISE provides:

- Online learning resources and academic skills workshops http://www.students.mq.edu.au/support/learning_skills/
- Personal assistance with your learning & study related questions.
- The Learning Help Desk is located in the Library foyer (level 2).
- Online and on-campus orientation events run by Mentors@Macquarie.

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

Details of these services can be accessed at http://www.student.mq.edu.au/ses/.

IT Help

If you wish to receive IT help, we would be glad to assist you at 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 and it outlines what can be done.

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

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs (from algorithms), showing an understanding of the underlying

architecture of the computer

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

Assessment tasks

- Weekly exercise
- Workshop participation
- Diagnostic Quiz
- Assignment 1
- Assignment 2
- Assignment 3
- Mid-term 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

- Understand the fundamental concept of algorithm and demonstrate enhanced problem solving/algorithmic skills
- Implement programs following standard software engineering practices (in particular document, test and debug)

Assessment tasks

- Weekly exercise
- Diagnostic Quiz
- Assignment 1
- Assignment 2
- Assignment 3
- Mid-term 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
- Workshop participation
- Assignment 1
- Assignment 2
- Assignment 3

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Assessment task

- Workshop participation

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:

	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.

	Show poor programming skills. Limited ability to write code that compiles or excutes 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.

	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).

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.

What has changed since last offering

Two new assessment items have been added

An online diagnostic quiz worth 2%

Workshop attendance and active participation worth 3%

Accordingly, the weight of the the three assignments is 5%, 10%, and 10% instead of 6%, 12%, and 12% in the past.

New version of Java

Because Java 6 is no longer supported, we have moved to Java 7, see the Delivery and Resources Section for more details