ITEC625

Fundamentals of Computer Science

S1 Evening 2015

Dept of Computing

Contents

General Information

Unit convenor and teaching staff
Convenor, lecturer
Scott McCallum
scott.mccallum@mq.edu.au
Contact via By email or call 9850 9575
E6A 375
By appointment

Lecturer
Gaurav Gupta
gaurav.gupta@mq.edu.au
Contact via By email or call 9850 6341
E6A 330
By appointment

Credit points
4

Prerequisites

Corequisites

Co-badged status
COMP125

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)

Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

General Assessment Information

Late Submission

Late submission of assignments will not be accepted, except in the event of unavoidable disruption.

If you experience unavoidable disruption, and wish to apply for late submission, please submit a Disruption to Studies request with appropriate evidence.

Assessment Tasks

Name	Weighting	Due
Weekly exercise	11%	End of every week
Assignment 1	8%	Sun 29 March, 11:45 pm
Assignment 2	8%	Sun 03 May, 11:45 pm
Assignment 3	8%	Sun 31 May, 11:45 pm
Test 1	5%	Tues 24 March, 6 pm (week 5)
Test 2	5%	Tues 28 April, 6 pm (week 8)
Test 3	5%	Tues 19 May, 6 pm (week 11)
Final examination	40%	TBA
Algorithm design and analysis	10%	Week 12

Weekly exercise

Due: **End of every week**
Weighting: **11%**

Each week, an online quiz or a programming exercise will be made available online. Each task is worth 1 mark. A maximum of 11 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)

Assignment 1

Due: **Sun 29 March, 11:45 pm**
Weighting: **8%**

This will be a programming assignment that will require you to write Java code to meet a set of requirements. The task will help you to practise concepts from the first topic grouping (weeks 1-3). 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: **Sun 03 May, 11:45 pm**
Weighting: **8%**

This will be a programming assignment that will help you to practise concepts from the second topic grouping (weeks 4-6). You 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: **Sun 31 May, 11:45 pm**
Weighting: **8%**

This will be a programming assignment that will help you to practise concepts from the third topic grouping (weeks 7-9). 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)

Test 1

Due: **Tues 24 March, 6 pm (week 5)**
Weighting: **5%**

This will be a 25-minute written test held during lecture class time which will test your understanding of material from the first topic grouping (weeks 1-3). 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

Test 2

Due: **Tues 28 April, 6 pm (week 8)**
Weighting: **5%**

This will be a 25-minute written test held during lecture class time which will test your understanding of material from the second topic grouping (weeks 4-6). 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

Test 3

Due: **Tues 19 May, 6 pm (week 11)**
Weighting: **5%**

This will be a 25-minute written test held during lecture class time which will test your understanding of material from the third topic grouping (weeks 7-9). 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: **40%**

This will be a three hour written invigilated examination which will cover all four topic groupings (that is, weeks 1-12).

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

Algorithm design and analysis

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

Given a suitable problem suggested by the lecturer, you will be requested to design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm.

On successful completion you will be able to:

- Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

Delivery and Resources

CLASSES

Each week you should attend

- three hours of lectures and
- a two hour mixed workshop class, consisting of a one hour tutorial followed by a consultation hour in the same lab.

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 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.

HELP101

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

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

Textbook

The first book in the following list, namely Savitch, is the required text book for COMP125. The other books listed are helpful references.

- W. Savitch, Absolute Java (Pearson) 5th edition. ISBN 9780132830317 (covers basic Java programming and data structures, ideal for COMP125 and may be useful beyond COMP125, as well. 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

The twelve weekly topics will be grouped into four 3-week groupings:

1. Introductory programming with numbers, strings and arrays, comprising weeks 1-3.
2. Programmer defined objects and object arrays, comprising weeks 4-6.
3. Indexed lists and recursion, comprising weeks 7-9.
4. Linked lists etc., comprising weeks 10-12.

The first, second and third topic groupings will each have an in-lecture-class test and an assignment, as detailed under `Assessment Tasks'. For the fourth topic grouping there will be a test paper provided purely for self assessment (that is, carrying no weight, and not to be marked by staff). The compulsory final examination will cover all topic groupings (for details see `Assessment Tasks').

1	Introducing Java	Gaurav
2	Developing and Testing Java Programs, Introducing Arrays	Gaurav
3	More on Arrays and Simple Searching	Gaurav
4	Classes and Objects	Gaurav
5	Arrays of Objects, More on Searching	Gaurav
6	Sorting Algorithms	Gaurav
	Recess	
7	Exception Handling	Scott
8	Indexed Lists (and the ArrayList class)	Scott
9	Recursive Algorithms	Scott
10	Linked Lists	Scott
11	Stacks and Queues	Scott
12	Object Oriented Design & Development	Scott
13	Review	Gaurav + 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

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

Disruption to Studies Policy 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/

Results

Results shown in *iLearn*, 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.

Disruption to Studies

The University recognises that students may experience disruption that adversely affects their academic performance in assessment activities. Support services are provided by the University, and it is the student's responsibility to access such services as appropriate. For information concerning the Disruption policy and procedures please visit http://students.mq.edu.au/student_admin/exams/disruption_to_studies/

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 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

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

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:

Learning outcome

- Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

Assessment tasks

- Weekly exercise
- Assignment 1
- Assignment 2
- Assignment 3
- Algorithm design and analysis

Capable of Professional and Personal Judgement and Initiative

We want our graduates to have emotional intelligence and sound interpersonal skills and to demonstrate discernment and common sense in their professional and personal judgement. They will exercise initiative as needed. They will be capable of risk assessment, and be able to handle ambiguity and complexity, enabling them to be adaptable in diverse and changing environments.

This graduate capability is supported by:

Assessment task

- Final examination

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)
- Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

Assessment tasks

- Weekly exercise
- Assignment 1
- Assignment 2
- Assignment 3
- Test 1
- Test 2
- Test 3
- Final examination
- Algorithm design and analysis

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcome

- Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

Assessment tasks

- Final examination
- Algorithm design and analysis

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)
- Given a suitable problem, design an algorithm to solve the problem, prove that your algorithm correctly solves the problem, and analyze the time complexity of your algorithm

Assessment tasks

- Weekly exercise
- Assignment 1
- Assignment 2
- Assignment 3
- Test 1
- Test 2
- Test 3
- Final examination
- Algorithm design and analysis

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

- Final examination

Changes from Previous Offering

We will (again) pay attention to the transition from COMP115 and Processing to the use of Java and Eclipse in COMP125. This year COMP125 will be weakly modularized into four 3-week topic groupings. For Session 1, there will be three in-class tests. Test 1 will cover the first topic grouping (weeks 1-3), Test 2 will cover the second topic grouping (weeks 4-6), and Test 3 will cover the third topic grouping (weeks 7-9). These tests will be marked and feedback provided. Test 4, covering the fourth topic grouping, will be offered purely for self-assessment, has 0% weighting, and will not be marked by staff. We will retain a compulsory final examination which will cover all four topic groupings (that is, weeks 1-12). Our hope is that these modifications will give students more practice with the later topics of this unit, and hence will help them to be better prepared for their final examination.

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 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.
L.O. #3	Inability 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.

L.O. #4	Poor ability to design an algorithm. Inability to prove that an algorithm correctly solves a problem. Inability to analyze the time complexity of an algorithm.	Reasonable competence in algorithm design. Basic capability to prove that an algorithm is correct. Adequate ability to analyze an algorithm.	Demonstrate superior ingenuity or insight in algorithm design. Show proficiency in algorithm proof. Demonstrate superior ability to analyze an algorithm.

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 final examination.

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.