



COMP225

Algorithms and Data Structures

S1 Day 2014

Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	3
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	7
<u>Unit Schedule</u>	10
<u>Policies and Procedures</u>	10
<u>Graduate Capabilities</u>	12
<u>Changes from last offering</u>	16

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

Oldooz Dianat

Oldooz.Dianat@mq.edu.au

Contact via Oldooz.Dianat@mq.edu.au

Tutor

Lee Coady

lee.coady@mq.edu.au

Contact via lee.coady@mq.edu.au

E6A 369

Unit Convenor

Annabelle McIver

annabelle.mciver@mq.edu.au

Contact via annabelle.mciver@mq.edu.au

Wednesday 3pm--4pm; other times by appointment.

Lecturer

Luke Mathieson

luke.mathieson@mq.edu.au

Contact via luke.mathieson@mq.edu.au

Credit points

3

Prerequisites

(COMP125(P) or COMP165(P)) and (3cp(P) from MATH132-MATH136 or DMTH137)

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://www.mq.edu.au/study/calendar-of-dates>

Learning Outcomes

On successful completion of this unit, you will be able to:

Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.

Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.

Be able to apply commonly used data structures (including trees, graphs, lists and their variations).

Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification

Have an understanding of the importance of documentation, testing, readability and modularity of programs

Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)

Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)

The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment Tasks

Name	Weighting	Due
Diagnostic Assessment	4%	Week 4
Assignment 1	8%	Week 7
Assignment 2	8%	Week 11
Portfolio	25%	Week 13
Weekly class activities	5%	Weekly
Formal Examination	50%	Examination period

Diagnostic Assessment

Due: **Week 4**

Weighting: **4%**

This assignment will focus on the relationship between algorithm design and performance. You will be asked to identify a number of sorting algorithms using systematic testing. You will also need to write some small Java programs to set up and collect test data.

The goal of the Diagnostic Task is to give you early feedback on how well you are drawing together knowledge from previous units in order to tackle the problems investigated in this unit.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)

Assignment 1

Due: **Week 7**

Weighting: **8%**

In this assignment you will be asked to design and analyse an algorithm based on Tree data structures. Your algorithm will be implemented in the Java programming language using some of the design techniques taught in lectures and the weekly exercises. The focus is on correctness and the ability to use recursion on tree data structures.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification
- Have an understanding of the importance of documentation, testing, readability and modularity of programs
- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)

Assignment 2

Due: **Week 11**

Weighting: **8%**

You will be asked to design and implement an algorithm in Java based on graph data structures using some of the more advanced techniques discussed in lectures.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification
- Have an understanding of the importance of documentation, testing, readability and modularity of programs
- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)

Portfolio

Due: **Week 13**

Weighting: **25%**

Based on the weekly class activities, and the assignments you will be asked to compile a portfolio of achievements in three areas:

- Algorithm design and correctness
- Performance and resource awareness of algorithms
- Collaboration, working methods and time management

The portfolio can be viewed as something you might present to a potential employer, including examples of your code, your comments and your reports to give concrete evidence that you understand the topics covered in the unit. There will be two opportunities in the semester to hand your portfolio in to obtain feedback and help in compiling it. The final portfolio is due in by week 13.

Final submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Have an understanding of the importance of documentation, testing, readability and modularity of programs
- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)
- Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Weekly class activities

Due: **Weekly**

Weighting: **5%**

Each week you will be asked to complete a selection of group-based activities in class. Each group will make a single submission which will be marked by the class tutor. Marks allocated will be based on the level of participation within the group as well as the quality of the submitted solution.

Submission method: via iLearn.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification
- Have an understanding of the importance of documentation, testing, readability and modularity of programs
- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)

- Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Formal Examination

Due: **Examination period**

Weighting: **50%**

A formal written examination based on lectures, class work and activities and assignments.

Submission method: Formal examination organised by the University.

On successful completion you will be able to:

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)

Delivery and Resources

Technology required

- [Eclipse](#) - download Eclipse IDE for Java Developers: The practical work in this unit involves programming in Java (www.java.com) using the Eclipse Integrated Development Environment (www.eclipse.org)
- [Java SE JDK](#) - download Java SE 6 (not 7) to be compatible with the labs: Note that you need the Java JDK which includes the compiler tools, rather than the Java Runtime Environment (JRE) which you might already have installed on your computer to allow you to run Java applications.
- Learning Management System [iLearn](#) : This will be used primarily to enable email broadcasts and give access to Assessment marks.

- Echo (formerly iLecture): the lecture audio will be recorded, and will be available via iLearn and Open Learning.

Classes

Each week you should attend **3 hours of lectures** and **a two-hour mixed classes**. For details of days, times and rooms consult the [timetables webpage](#).

You should have selected one two-hour mixed classes session at enrolment. **You must attend the session you are enrolled in.**

The class exercises are intended to be completed in small groups of around four participants. In the first class you will be assigned a group. You will need to work with the group members for the whole term. A single group submission for the weekly exercises will be expected, and in the following week your tutor will supply feedback and expect the group participants to answer questions about the submission. The 5% course credit for the group work will depend on both the quality of the submitted answers as well as the level of participation.

Please note that you are **expected** to attend most of the mixed classes and **required** to submit work by the stated deadlines. Failure to do so may result in you failing the unit or being excluded from the exam.

Required and Recommended Texts

The textbook for Java programming used this semester is:

Adam Drozdek [2005]. *Data Structures and Algorithms in Java* (2nd ed. or 3rd edition). Boston: Thomson Course Technology.

This textbook is available from the University Co-op Bookshop.

There is also a [companion website](#) by the publisher, containing data files for exercises. In addition, Drozdek has Java code from the book available on his [webpage](#). (Note that these are written for Java 1.4.)

Unit Web Page

The web page for this unit can be found at <http://www.comp.mq.edu.au/units/comp225/>. Note that the majority of the unit materials are on Open Learning.

The unit will make use of discussions hosted within iLearn. Please post questions there, they will be monitored by the staff on the unit.

Teaching and Learning Strategy

COMP225 is taught via lectures and mixed classes in the laboratory. Lectures are used to introduce new theoretic material, give examples of the use these techniques and put them in a wider context. Mixed classes give you the opportunity to interact with your peers. You will be given problems to solve each week prior to each session; preparing solutions is important because it will allow you to discuss the problems effectively with your tutor and maximise the feedback you get on your work. The aim of the mixed classes is to help you to develop problem-

solving skills and teamwork, and you will be expected to work on problems in class. Mixed classes give you an opportunity to practice your programming skills, and to implement many of the ideas discussed in lectures. Each week you will be given a number of problems to work on; it is important that you keep up with these problems as doing so will help you understand the material in the unit and prepare you for the work in assignments. Some of the questions are designated **compulsory** and they contribute to your total grade via the weekly exercise assessment and your portfolio. These must be completed and submitted by the deadline.

Lecture notes will be made available each week but these notes are intended as an outline of the lecture only and are not a substitute for your own notes or the textbook.

Standards and Grading

Your **final grade** will depend on your overall performance, **but note that** your performance in each part separately will also be taken into account.

To be eligible for **special consideration** for the final examination you must have been performing satisfactorily in the unit prior to any unavoidable disruption.

Extension requests: Please note that if you repeatedly hand in work late then this will impact your final grading of the portfolio. If you cannot submit on time because of illness or other circumstances, please contact the lecturer **before** the due date.

Summary of achievement required corresponding to each final grade

- HD and D

Overall the quality of the work demonstrates a mature and considered appreciation of the programming and algorithmic concepts, and an excellent technical mastery of Java programming (sufficient to complete the advanced programming tasks). A systematic demonstration of the ability to problem solve independently and a thorough knowledge of how to critique the proposed solution, in terms of performance, correctness and other technical issues.

- Cr

Overall the quality of the work demonstrates a reasonable appreciation of the programming and algorithmic concepts, and a good technical mastery of Java programming (sufficient to complete the required programming tasks). A systematic demonstration of the ability to solve basic problems and to present the solutions clearly with an attempt to give reasons why they meet their stated objectives. Some knowledge of how to critique the proposed solution, in terms of performance, correctness and other technical issues is demonstrated, but the answers given might not cover all cases.

- P The quality of work demonstrates a basic technical mastery of the Java language, a basic understanding of how to program using the studied algorithms and a knowledge of how to implement and use the basic algorithmic data structures and programming techniques introduced in the course. The assessment work demonstrates a basic understanding of performance and correctness issues relative to all of the algorithms and data structures studied in the unit, and the appropriateness of a particular algorithm relative to a given data structure.

Unit Schedule

Week 1	Review of algorithms and related concepts
Week 2	Algorithm Correctness and Efficiency
Week 3	Algorithm Design Strategies
Week 4	Sorting
Week 5	Binary Trees
Week 6	Binary Trees (cont.)
April 14--25	Mid-Semester Break
Week 7	Priority Queues, Heaps and Heapsort
Week 8	Programming with Maps and Hashtables
Week 9	Graph Algorithms
Week 10	Graph Algorithms (cont.)
Week 11	Advanced Trees
Week 12	An Introduction to Computability
Week 13	Revision

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/

Department of Computing special Consideration: http://comp.mq.edu.au/undergrad/policies/special_consideration_policy.htm

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

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:

Learning outcomes

- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment tasks

- Portfolio
- Weekly class activities

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

Learning outcomes

- Time management in meeting assignment and weekly exercise deadlines. The ability to read, analyse and understand written material (especially technical documentation)
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

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

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).

Assessment tasks

- Assignment 1
- Assignment 2
- Formal Examination

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 outcomes

- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification

Assessment tasks

- Diagnostic Assessment
- Assignment 1
- Assignment 2
- Portfolio
- Weekly class activities

- Formal 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

- Be familiar with a variety of algorithm design techniques and understand how they can improve either efficiency or clarity.
- Be aware of the importance of correctness for algorithms, and able to apply strategies for achieving correctness.
- Be able to apply commonly used data structures (including trees, graphs, lists and their variations).
- Have an understanding of the importance of documentation, testing, readability and modularity of programs

Assessment tasks

- Diagnostic Assessment
- Assignment 1
- Assignment 2
- Weekly class activities
- Formal 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:

Learning outcomes

- Carrying out advanced and broadly based problem solving - most particularly when designing and writing programs in Java to meet a given specification
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment tasks

- Diagnostic Assessment
- Assignment 1
- Assignment 2
- Portfolio

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:

Learning outcomes

- Have an understanding of the importance of documentation, testing, readability and modularity of programs
- Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment tasks

- Portfolio
- Weekly class activities
- Formal Examination

Engaged and Ethical Local and Global citizens

As local citizens our graduates will be aware of indigenous perspectives and of the nation's historical context. They will be engaged with the challenges of contemporary society and with knowledge and ideas. We want our graduates to have respect for diversity, to be open-minded, sensitive to others and inclusive, and to be open to other cultures and perspectives: they should have a level of cultural literacy. Our graduates should be aware of disadvantage and social justice, and be willing to participate to help create a wiser and better society.

This graduate capability is supported by:

Learning outcomes

- Basic written and oral communication skills (writing technical documents and program

documentation, communicating problems and their solutions in small and medium sized group situations)

- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment tasks

- Portfolio
- Formal Examination

Socially and Environmentally Active and Responsible

We want our graduates to be aware of and have respect for self and others; to be able to work with others as a leader and a team player; to have a sense of connectedness with others and country; and to have a sense of mutual obligation. Our graduates should be informed and active participants in moving society towards sustainability.

This graduate capability is supported by:

Learning outcomes

- Basic written and oral communication skills (writing technical documents and program documentation, communicating problems and their solutions in small and medium sized group situations)
- The ability to contribute to self and others' learning through constructive comments within the learning environment.

Assessment tasks

- Portfolio
- Weekly class activities

Changes from last offering

Although COMP225 uses a portfolio as a major assessment task through which students have an opportunity to demonstrate their learning with respect to the graduate capabilities, we are reintroducing credit for weekly exercises. This is different to last year's offering and the reason for it is to improve group work and communication in the learning environment.