



MONASH University

FIT9015
Data structures and algorithms

Unit Guide

Semester 2, 2009

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated : 16 Jul 2009

Table of Contents

<u>FIT9015 Data structures and algorithms - Semester 2, 2009</u>	1
<u>Chief Examiner:</u>	1
<u>Lecturer(s) / Leader(s):</u>	1
<u>Gippsland</u>	1
<u>Introduction</u>	2
<u>Unit synopsis</u>	2
<u>Learning outcomes</u>	2
<u>Contact hours</u>	3
<u>Workload</u>	3
<u>Unit relationships</u>	3
<u>Prerequisites</u>	3
<u>Prohibitions</u>	3
<u>Relationships</u>	3
<u>Teaching and learning method</u>	4
<u>Timetable information</u>	4
<u>Tutorial allocation</u>	4
<u>Unit Schedule</u>	4
<u>Unit Resources</u>	5
<u>Prescribed text(s) and readings</u>	5
<u>Recommended text(s) and readings</u>	5
<u>Required software and/or hardware</u>	5
<u>Equipment and consumables required or provided</u>	5
<u>Study resources</u>	5
<u>Assessment</u>	7
<u>Overview</u>	7
<u>Faculty assessment policy</u>	7
<u>Assignment tasks</u>	7
<u>Examination</u>	8
<u>Due dates and extensions</u>	8
<u>Late assignment</u>	8
<u>Return dates</u>	9
<u>Appendix</u>	10

FIT9015 Data structures and algorithms - Semester 2, 2009

Chief Examiner:

Associate Professor Manzur Murshed

Head of School

Phone: +61 3 990 26467

Fax: +61 3 990 26842

Lecturer(s) / Leader(s):

Gippsland

Associate Professor Manzur Murshed

Head of School

Phone: +61 3 990 26467

Fax: +61 3 990 26842

Introduction

Welcome to FIT9015 Data Structures and Algorithms for Semester 2, 2009. This level-9 unit has been designed to provide you understanding of how data is stored to achieve higher efficiency while developing algorithms to solve programming problems in the IT profession. It explores wide range of data structures and algorithms with emphasis on applying the knowledge into programming applications.

Unit synopsis

Algorithm analysis. Application and implementation of some common data structures: stacks, queues, lists, priority queues, tables, sets and collections. Data representations including: arrays, linked lists, heaps, trees (including balanced trees) and hashing. Design of application programs making use of common data structures. Design and implementation of new data structures. Study of advanced algorithms in areas such as: graph theory, pattern searching and data compression. Access to the University's computer systems through an Internet service provider is compulsory for off-campus students

Learning outcomes

1. Ability to analyse simple algorithms to work out an order of magnitude estimate of running time and space
2. Familiarity with some of the most common data structures:

- stacks
- queues
- lists
- priority queues
- tables
- sets
- collections

Ability to implement these data structures using various common data representations:

- arrays
- linked lists
- heaps
- trees (including balanced trees)
- hashing

Ability to evaluate which implementation would be most appropriate for a given data structure and application. Ability to apply the same principles used in implementing the common data structures to implement other data structures. Ability to design and implement new data structures. Understanding of some more advanced algorithms in areas such as:

- graph theory (shortest path etc)
- pattern searching
- data compression
- (precise selection of advanced algorithms will vary from year to year)

Ability to design new algorithms to solve new problems. Enjoyment of programming as an intellectual exercise. Appreciation of the elegance of certain data structures and algorithms as a form of art. Interest in understanding how data structures and algorithms are implemented rather than merely using other people's implementations (and consequently a preference for open source software).

Contact hours

One x 2 hr lecture/week

One x 2 hr laboratory (programming session in PC lab)/week. (Depending on on-campus class size a repeat laboratory may be required.)

Workload

For on campus students, workload commitments are:

- two-hour lecture;
- two-hour tutorial/laboratory (requiring advance preparation); and
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

You will need to allocate up to 5 hours per week in some weeks, for use of a computer, including time for newsgroups/discussion groups.

Off-campus students generally do not attend lecture and tutorial sessions, however, you should plan to spend equivalent time working through the relevant resources and participating in discussion groups each week.

Unit relationships

Prerequisites

FIT1007 or GCO1812 or FIT9013 or GCO9808

Prohibitions

CSE2304, FIT2009, GCO2817, GCO3512, and GCO9807 (translation set)

Relationships

FIT9015 is a level-9 unit of the Master of Applied Information Technology (MAIT) degree.

Teaching and learning method

The approach to teaching and learning include a weekly two-hour lecture and a two-hour (tutorial/laboratory). Additionally, each student should spend a minimum of 8 to 12 hours for personal study every week and should allocate up to 5 hours per week in some weeks for use of a computer, including time for newsgroup and discussion.

Timetable information

For information on timetabling for on-campus classes please refer to MUTTS, <http://mutts.monash.edu.au/MUTTS/>

Tutorial allocation

On-campus students should register for tutorials/laboratories using the Allocate+ system:
<http://allocate.cc.monash.edu.au/>

Unit Schedule

Week	Topic	Study guide	Key dates
1	Generic Data Structures	1	
2	Algorithm Analysis	2	
3	Developing Algorithms	3	
4	Sorting Algorithms	4	
5	Lists	5	
6	Stacks and Queues	6	
7	Graphs and Trees	7	Assignment 1 due
8	Binary Search Trees	8	
9	Hashing	9	
10	Heaps	10	
Mid semester break			
11	Some Applications of Data Structures	11	Assignment 2 due
12	Revision		

Unit Resources

Prescribed text(s) and readings

Mark Allen Weiss, *Data Structures & Problem Solving using Java*, 3rd Edition, Addison Wesley, 2006, ISBN: 0-321-31255-4.

Text books are available from the Monash University Book Shops. Availability from other suppliers cannot be assured. The Bookshop orders texts in specifically for this unit. You are advised to purchase your text book early.

Recommended text(s) and readings

William H. Ford and William R. Topp, *Data Structures with Java*, 2005, Pearson Education International, ISBN 0131293370

Lafore, R., *Data Structures & Algorithms in Java*, 2nd edition, 2002, SAMS, ISBN 0-672-32453-9

Robert Sedgewick and Michael Schidlowsky, *Algorithms in Java*, 3rd edition (Parts 1-4), Addison Wesley, 2002, ISBN:0201361205.

Mitchell Waite and Robert Lafore, *Data Structures & Algorithms in Java*, Waite Group Press, 1998, ISBN: 1571690956.

Donald Ervin Knuth, *Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3rd edition, Addison Wesley, 1997, ISBN:0201896834.

Donald Ervin Knuth, *Art of Computer Programming, Volume 3: Sorting and Searching*, 2nd edition, Addison Wesley, 1998, ISBN:0201896850.

Required software and/or hardware

Java SE JDK version 1.5 (also known as version 5) or later.

This software is included in the GSIT Unit Software CD-ROM, which will be sent to all students.

The software may also be downloaded free from <http://java.sun.com>

Equipment and consumables required or provided

Students studying off-campus are required to have the minimum system configuration specified by the Faculty as a condition of accepting admission, and regular Internet access. On-campus students, and those studying at supported study locations may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to 8 hours per week for use of a computer, including time for newsgroups/discussion groups.

Study resources

Study resources we will provide for your study are:

- A Unit Book containing 11 study guides at MUSO.
- This Unit Information outlining the administrative information for the unit.

- A CD-ROM (possibly sent at the start of the year if you were enrolled in a first semester unit that required it) with software required for all units (this includes all the software required to complete this unit).
- A unit web page at MUSO where lecture slides, weekly tutorial requirements, assignment specifications, sample solutions and supplementary material will be posted.
- Discussion groups at MUSO.

Assessment

Overview

Assignments: 40%

Examination: 60%

Faculty assessment policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 44% then a mark of no greater than 44-N will be recorded for the unit.

The unit is assessed with two assignments and a three hour closed book examination. To pass the unit you must:

- attempt both assignments and the examination
- achieve no less than 40% of the possible average marks in the two assignments
- achieve no less than 40% of the possible marks in the exam
- achieve no less than 50% of possible marks

Assignment tasks

Assignment coversheets

Assignment coversheets are available via "Student Forms" on the Faculty website:

<http://www.infotech.monash.edu.au/resources/student/forms/>

You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

Assignment submission and return procedures, and assessment criteria will be specified with each assignment.

• Assignment task 1

Title:

Assignment 1

Description:

Students will be required to perform a number of tasks involving both analytical and practical (computer programming) skills from the syllabus covered in Study Guides 1-4.

Weighting:

20%

Due date:

31 August 2009

• Assignment task 2

Title:

Assignment 2

Description:

Students will be required to perform a number of tasks involving both analytical and practical (computer programming) skills from the syllabus covered in Study Guides 5-8.

Weighting:

20%

Due date:

5 October 2009

Examination

- **Weighting:** 60%

Length: 3 hours

Type (open/closed book): Closed book

See Appendix for End of semester special consideration / deferred exams process.

Due dates and extensions

Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are not regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Students requesting an extension for any assessment during semester (eg. Assignments, tests or presentations) are required to submit a Special Consideration application form (in-semester exam/assessment task), along with original copies of supporting documentation, directly to their lecturer within two working days before the assessment submission deadline. Lecturers will provide specific outcomes directly to students via email within 2 working days. The lecturer reserves the right to refuse late applications.

A copy of the email or other written communication of an extension must be attached to the assignment submission.

Refer to the Faculty Special consideration webpage or further details and to access application forms:

<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>

Late assignment

- An assignment must be submitted by the *cut-off* date, which is usually seven days after the due date. Any assignment submitted after the cut-off date will not be accepted by the MUSO system and therefore, it will be marked automatically to zero.
- Any assignment submitted after the due date will be penalised by 5% of the full marks for each 24 hours of delay.
- This policy is strict because comments or guidance will be given on assignments as they are returned, and sample solutions may also be published and distributed, after assignment marking or with the returned assignment.

Return dates

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

Appendix

Please visit the following URL: <http://www.infotech.monash.edu.au/units/appendix.html> for further information about:

- Continuous improvement
- Unit evaluations
- Communication, participation and feedback
- Library access
- Monash University Studies Online (MUSO)
- Plagiarism, cheating and collusion
- Register of counselling about plagiarism
- Non-discriminatory language
- Students with disability
- End of semester special consideration / deferred exams