MONASH University
Information Technology

FIT1008
Introduction to computer science

Unit Guide

Semester 1, 2013

Last updated: 04 Mar 2013

Table of Contents

FIT1008 Introduction to computer science - Semester 1, 2013 1
- Mode of Delivery 1
- Contact Hours 1
- Workload requirements 1
- Unit Relationships 1
  - Prohibitions 1
  - Prerequisites 1

Chief Examiner 1
Campus Lecturer 1
- Clayton 2

Tutors 2
- Clayton 2

Academic Overview 3
- Learning Outcomes 3

Unit Schedule 4
- Assessment Summary 4
- Teaching Approach 5

Assessment Requirements 6
- Assessment Policy 6
- Assessment Tasks 6
  - Hurdle Requirements 6
  - Participation 6

Examinations 7
- Examination 1 7

Learning resources 7
Reading list 7
Feedback to you 8
Extensions and penalties 8
Returning assignments 8
Assignment submission 8
Online submission 9
Required Resources 9

Other Information 10
- Policies 10
  - Graduate Attributes Policy 10

Student services 10
Monash University Library 10
Disability Liaison Unit 11
Your feedback to Us 11
Previous Student Evaluations of this Unit 11

FIT1008 Introduction to computer science - Semester 1, 2013

This unit introduces students to core problem-solving, analytical skills, and methodologies useful for developing flexible, robust, and maintainable software. In doing this it covers a range of conceptual levels, from high level algorithms and data-structures, down to abstract machine models and simple assembly language programming. Topics include data structures; algorithms; object-oriented design and programming; and abstract machines.

Mode of Delivery

Clayton (Day)

Contact Hours

3 hrs lectures/wk, 3 hrs laboratories/wk, 1 hr tutorial/wk

Workload requirements

Students will be expected to spend a total of 12 hours per week during semester on this unit as follows:

- Lectures: 3 hours per week
- Tutorial: 1 hour per week
- Computer Lab Prac: 1.5 hours per week (requiring advance preparation) followed by an extra 1.5 hours for prac marking

- a minimum of 5 hours of personal study per week in order to satisfy the reading, assignment expectations, including time for newsgroups/discussion groups.

Unit Relationships

Prohibitions

CSE1303, CSC1030, FIT1007, FIT1015

Prerequisites

(FIT1040 or FIT1002) and FIT1029

Chief Examiner

Professor Maria Garcia de la Banda

Campus Lecturer

Clayton

Aldeida Aleti

Consultation hours: Tuesday 11am - 1pm, Room 133a, Building 63

Tutors

Clayton

Aldeida Aleti

Academic Overview

Learning Outcomes

At the completion of this unit, students will haveDeveloped the ability to:

- understand abstract data types and, in particular, data structures for stacks, queues, lists, and trees, as well as their associated algorithms for creating and manipulating them. Evaluate the appropriateness of different data structures for a given problem;
- understand basic searching and sorting algorithms and implement them. Understand the concept of algorithmic complexity. Analyse the complexity of these searching and sorting algorithms as well as other basic algorithms. Compare the complexity of different algorithms for solving a given problem;
- analyse different implementations of abstract data types and determine their implications regarding complexity, functionality, and memory usage;
- understand the uses of recursive algorithms and data structures, their advantages and disadvantages. Analyse the complexity of simple recursive algorithms, and their relationship with iteration. Understand basic recursive algorithms for lists and trees, and develop new ones;
- understand the basic concepts in the object-oriented (OO) programming paradigm;
- understand the basic concepts in testing;
- understand the requirements for "good programming practice";
- understand the different compilation targets, including abstract machine code, assembly language, object code, and machine code. Understand the relationship between simple code in a high level imperative language and its low level translation into assembly code;
- learn the structure and design of a particular processor simulator. Analyse the execution in this simulator of simple iterative algorithms learned before, thus gaining a deeper understanding of the connection between software and hardware, between an algorithm and its execution;
- understand how the simulator implements function calling, and use it to reinforce the connection between recursion and iteration.

Developed attitudes that enable them to:

- conform to programming standards when writing software;
- use good design principles when constructing systems;
- take a patient and thorough approach to testing;
- acknowledge any assistance they have received in writing a program;
- search for information in appropriate places when necessary.

Developed the skills to:

- implement their own data-structures. Design and implement Java programs using a variety of data structures and algorithms;
- implement an object-oriented program consisting of several interacting classes requiring not only basic but also advance object-oriented concepts;
- construct a test harness for testing an object-oriented program;
- debug and modify an existing program (written by somebody else);
- use the Java API classes as part of their programs.

Demonstrated the communication skills necessary to:

- document a program correctly;
- explain how parts of a program work.

Unit Schedule

Week	Activities	Assessment
0	Register for tutorials, pracs and lectures	No formal assessment or activities are undertaken in week 0
1	Lectures on Revision of BigO and List (arrays). Tute & Prac.	Prac 1
2	Lectures on List Sorting & Stacks and Queues (arrays). Tute & Prac.	Prac 2
3	Lectures on Lists, Stacks and Queues (linked nodes). Tute & Prac	Prac 3
4	Lectures on Basic Object Oriented Programming. Tute & Prac.	Prac 4
5	Lectures on Advanced Object Oriented Programming. Tute & Prac.	Prac 5
6	Lectures on Recursion and Recursive Sorts. Tute & Prac.	Prac 6
7	Lectures on Binary Trees. Tute & Mid-semester Test. No Prac.	Mid-semester Test in Tuesday lecture
8	Lectures on Priority Queues and Heaps. Tute & Prac.	Prac 7
9	Lectures on Hash Tables & Number Representation. Tute & Prac.	Prac 8
10	Lectures on Computer Systems & MIPS. Tute & Prac	Prac 9
11	Lectures on MIPS execution of selection and iteration. Tute & Prac.	Prac 10
12	Lectures on MIPS function call/return. Tute & Prac	Prac 11
	SWOT VAC	No formal assessment is undertaken in SWOT VAC
	Examination period	LINK to Assessment Policy: http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html

*Unit Schedule details will be maintained and communicated to you via your learning system.

Assessment Summary

Examination (3 hours): 70%; In-semester assessment: 30%

Assessment Task	Value	Due Date
Mid-semester Test (1 hour)	10%	Week 7 in Tuesday lecture
Pracs (1.5 hours each)	20%	Weekly except in Week 7
Examination 1	70%	To be advised

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach helps students first encounter the information at lectures, discuss and explore them at length during tutorials, and practice them in a hands-on environment during labs.

Assessment Requirements

Assessment Policy

Faculty Policy - Unit Assessment Hurdles
(http://www.infotech.monash.edu.au/resources/staff/edgov/policies/assessment-examinations/unit-assessment-hu

Academic Integrity - Please see the Demystifying Citing and Referencing tutorial at
http://lib.monash.edu/tutorials/citing/

Assessment Tasks

Hurdle Requirements

There are four hurdles in this unit:

- First, students must actively participate in at least 7 out of 12 tutorials. Active participation includes contributing opinions to a discussion, providing an answer to some question/exercise, or posing a unit-related question.
- Second, students must achieve at least 50% of the total prac marks.
- Third, students must achieve at least 50% of the exam mark.
- Fourth, students must achieve at least 50% of the total marks.

Students who do not meet all these hurdles can get a maximum of 49N for the unit.

Participation

Students must actively participate in at least 7 out of the 12 tutorials. Active participation includes contributing opinions to a discussion, providing an answer to some question/exercise, or posing a unit-related question.

- **Assessment task 1**

  Title:
  Mid-semester Test (1 hour)
  Description:
  This test is performed during the class and covers exam-like questions for the first part of the course. It is intended to give students an idea of how they would perform in the final exam, given their current progress.
  Weighting:
  10%
  Criteria for assessment:
  This test will evaluate your understanding of the material provided during the first few weeks of semester, your capability to code simple algorithms given a clear specification, and to analyse the behaviour and complexity of simple fragments of code.
  Due date:
  Week 7 in Tuesday lecture

- **Assessment task 2**

Title:
Pracs (1.5 hours each)
Description:
Each week you will need to complete a prac assignment. In some of the pracs, you will need to work together with another student, but will you will be marked individually. Prac assignments are long (they are designed to take about 4 hours). This means that you must have a significant proportion of the prac completed before attending the scheduled computer lab. The aim of the 1.5 hour computer lab practical is to iron out any bugs, ask any questions about the prac you have not been able to solve on your own, and improve the parts that your demonstrator points out as lacking (including comments, algorithms, etc). If you do nothing before the 1.5 hours scheduled, you will soon realise that you do not have enough time to complete it. The prac sheets will be released every Thursday morning and made available on the unit's web page.
Weighting:
20%
Criteria for assessment:
Every prac sheet contains the assessment criteria used to assess that prac. In addition, demonstrators carry with them a marking guide prepared by the lecturer which indicates how exactly to mark each prac question. You can request the demonstrator to show you the marking guide after he/she has marked your prac.
Due date:
Weekly except in Week 7

Examinations

- **Examination 1**

Weighting:
70%
Length:
3 hours
Type (open/closed book):
Closed book
Electronic devices allowed in the exam:
None

Learning resources

Reading list

(1) *Data Structures and Algorithms in Java*. Robert Lafore, SAMS. This book provides a very simple approach to understanding data structures and algorithms. While the book uses Java to illustrate the implementation, its focus is on the actual data structures and algorithms, rather than on Java, which is very useful for first year students. Very basic and simple.

(2) *Data Structures and Algorithms in Java.* Adam Drozdek, Brooks/Cole. More advanced but still appropriate for average and high-end students.

(3) *Algorithms in Java*. Robert Sedgewick. Parts 1-4. This book is a more in-depth book and assumes a strong mathematical background from the reader. It is recommended for advanced students who want to

learn more about the complexity of the algorithms and data structures used.

(4) *Data Structures and Algorithm Analysis in Java*. Third Edition. Mark Allen Weiss. This is one of the books you will also use in FIT2004. It contains very clear discussions of the algorithms and data-structures, but it assumes knowledge of a difficult Java construct (Generics). Very recommended if you can get pass that.

(5) *The Java Programming Language.* Fourth Edition. Ken Arnold, James Gosling and David Holmes. Prentice Hall. This book is useful for students who have questions about the Java language. The reason for using the fourth edition is because it is the earliest edition that includes parametric polymorphism (called Generics in Java).

Monash Library Unit Reading List
http://readinglists.lib.monash.edu/index.html

Feedback to you

Types of feedback you can expect to receive in this unit are:

- Informal feedback on progress in labs/tutes
- Graded assignments without comments
- Test results and feedback
- Other: Detailed solutions to tutes

Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:
http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html.

Returning assignments

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

Assignment submission

It is a University requirement
(http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html) for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at http://www.infotech.monash.edu.au/resources/student/forms/. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an online quiz).

Online submission

Please submit your work via Moodle at the time in which you are marked (and, thus, **before** you leave the lab). You can access Moodle via links in the my.monash portal.

Required Resources

Please check with your lecturer before purchasing any Required Resources. Limited copies of prescribed texts are available for you to borrow in the library, and prescribed software is available in student labs.

Eclipse Platform. One of the two recommended platforms. It can be downloaded from http://www.eclipse.org/downloads/

NetBeans IDE. The other recommended platform. It can be downloaded fro http://netbeans.org/downloads/

Java Development Kit, Version j2sdk-1_5_0_06 or later, Sun Microsystems, Inc. You should download the freeware version as you will have no need for the fuller facilities provided in JCreatorPro, and would have to pay as well.

The MIPS R2000 simulator **SPIM S20** (with its new interface **QtSpim**).

All the above are included as part of the Standard Operating Environment used in Faculty Computer Labs.

Other Information

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at: www.policy.monash.edu.au/policy-bank/academic/education/index.html

Key educational policies include:

- Plagiarism; http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html
- Assessment in Coursework Programs; http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-po
- Special Consideration; http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.ht
- Grading Scale; http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html
- Discipline: Student Policy; http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html
- Academic Calendar and Semesters; http://www.monash.edu.au/students/dates/
- Orientation and Transition; http://intranet.monash.edu.au/infotech/resources/students/orientation/
- Academic and Administrative Complaints and Grievances Policy; http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy.h
- Code of Practice for Teaching and Learning; http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-teac

Graduate Attributes Policy

http://www.policy.monash.edu/policy-bank/academic/education/management/monash-graduate-attributes-policy.ht

Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at http://www.monash.edu.au/students. For Sunway see http://www.monash.edu.my/Student-services, and for South Africa see http://www.monash.ac.za/current/.

Monash University Library

The Monash University Library provides a range of services, resources and programs that enable you to save time and be more effective in your learning and research. Go to www.lib.monash.edu.au or the library tab in my.monash portal for more information. At Sunway, visit the Library and Learning Commons at http://www.lib.monash.edu.my/. At South Africa visit http://www.lib.monash.ac.za/.

Disability Liaison Unit

Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis.

Website: http://www.monash.edu/equity-diversity/disability/index.htmlTelephone: 03 9905 5704 to book an appointment with a DLO; or contact the Student Advisor, Student Commuity Services at 03 55146018 at SunwayEmail: dlu@monash.eduDrop In: Equity and Diversity Centre, Level 1, Building 55, Clayton Campus, or Student Community Services Department, Level 2, Building 2, Monash University, Sunway Campus

Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through the Student Evaluation of Teaching and Units (SETU) survey. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, see:

www.monash.edu.au/about/monash-directions and on student evaluations, see:
www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this Unit

As a result of student feedback quizzes have been added to each week to help students practice the concepts being taught.

If you wish to view how previous students rated this unit, please go to
https://emuapps.monash.edu.au/unitevaluations/index.jsp