



MONASH University
Information Technology

FIT1008
Introduction to computer science

Unit Guide

Semester 2, 2013

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: 25 Jul 2013

Table of Contents

<u>FIT1008 Introduction to computer science - Semester 2, 2013</u>	1
<u>Mode of Delivery</u>	1
<u>Contact Hours</u>	1
<u>Workload requirements</u>	1
<u>Unit Relationships</u>	1
<u>Prohibitions</u>	1
<u>Prerequisites</u>	1
<u>Chief Examiner</u>	1
<u>Campus Lecturer</u>	2
<u>Clayton</u>	2
<u>Sunway</u>	2
<u>Academic Overview</u>	3
<u>Learning Outcomes</u>	3
<u>Unit Schedule</u>	4
<u>Assessment Summary</u>	4
<u>Teaching Approach</u>	5
<u>Assessment Requirements</u>	6
<u>Assessment Policy</u>	6
<u>Assessment Tasks</u>	6
<u>Hurdle Requirements</u>	6
<u>Participation</u>	6
<u>Examinations</u>	7
<u>Examination 1</u>	7
<u>Learning resources</u>	7
<u>Reading list</u>	7
<u>Feedback to you</u>	8
<u>Extensions and penalties</u>	8
<u>Returning assignments</u>	8
<u>Assignment submission</u>	8
<u>Online submission</u>	9
<u>Required Resources</u>	9
<u>Other Information</u>	10
<u>Policies</u>	10
<u>Graduate Attributes Policy</u>	10
<u>Student services</u>	10
<u>Monash University Library</u>	10
<u>Disability Liaison Unit</u>	11
<u>Your feedback to Us</u>	11
<u>Previous Student Evaluations of this Unit</u>	11

FIT1008 Introduction to computer science - Semester 2, 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)
- Sunway (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, FIT1015

Students beginning FIT1008 are assumed to be able to: Identify the main components of an algorithm (variables, operators, expressions, etc), and write the algorithm associated to the specification of a simple problem. Be able to translate a simple algorithm into a program containing variable declarations, selection, repetition, and lists and/or arrays.

Prerequisites

(FIT1040 or ECE2071 or FIT1002) and FIT1029

Chief Examiner

Professor Maria Garcia de la Banda

Campus Lecturer

Clayton

Maria Garcia de la Banda

Consultation hours: TBC

Sunway

Jojo Wong

Consultation hours: TBC

Academic Overview

Learning Outcomes

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

- understand abstract data types and, in particular, data structures for stacks, queues, lists, trees, and heaps, as well as the associated algorithms for creating and manipulating them. Evaluate the appropriateness of different data structures for a given problem;
- understand basic searching and sorting algorithms, analyse their properties and be able to 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 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 programs using a variety of data structures and algorithms;
- construct a test harness for testing a program;
- debug and modify an existing program (written by somebody else);
- use an Integrated Development Environment when developing 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 Python Basics, Sorting Revision and Big O. Tute & Prac.	Prac 1
2	Lectures on Abstract Data Types, Arrays, Sorted/Unsorted Lists. Tute & Prac.	Prac 2
3	Lectures on Python Basics II, Stacks and Queues (arrays). Tute & Prac	Prac 3
4	Lectures on Python OO and Stacks and Queues (linked nodes). Tute & Prac.	Prac 4
5	Lectures on Linked Lists, Iterators and Higher Order. 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 Monday 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 Monday 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 Monday 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 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

For algorithms and data structures:

(1) Reed and Zelle - Data Structures and Algorithms using Python and C++

(2) Rance D. Nicaise - Data Structures and Algorithms using Python

For learning Python 3, beginners:

(1) How to think like a computer scientist:<http://openbookproject.net/thinkcs/python/english3e/>

Assessment Requirements

(2) Learn Python the hard way (BUT: it is for python 2, requires reworking the examples with the notes below): <http://learnpythonthehardway.org/book/>

(2.1) Notes for the conversion of the exercises to Python 3:
<http://archive.p2pu.org/node/16646/forums/25197>

(3) John Guttag -- Introduction to Computation and Programming using Python

For learning Python 3, more experienced programmers:

(1) Dive into Python 3: <http://getpython3.com/diveintopython3/>

(2) Python tutorial: <http://docs.python.org/3/tutorial/>

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.monash.edu.au/exams/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). Please note that it is your responsibility to retain copies of your assessments.

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.

Python 3: This will be our programming language. Please make sure you get version 3.3

Aptana: A good IDE. It's a packaging of eclipse + PyDev plugin + extras. Requires Java to run. It is very complete but a bit heavy. Since it runs on Java, it runs for any version of Python. <http://www.aptana.com>

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:

- Academic integrity;
<http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-policy.html>
- Assessment in Coursework Programs;
<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html>
- Special Consideration;
<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.html>
- 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.html>
- Code of Practice for Teaching and Learning;
<http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-teaching-and-learning.html>

Graduate Attributes Policy

<http://www.policy.monash.edu/policy-bank/academic/education/management/monash-graduate-attributes-policy.html>

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.html> Telephone: 03 9905 5704 to book an appointment with a DLO; or contact the Student Advisor, Student Community Services at 03 55146018 at Sunway Email: dlu@monash.edu Drop 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, we have done the following:

- Added programs with the code used in all lectures. This code gives you a good idea of what we want in terms of commenting and testing.
- Added quizzes 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>