MONASH University

FIT1015
Computer science

Unit guide

Semester 1, 2009

Last updated : 20 Apr 2009

Table of Contents

FIT1015 Computer science - Semester 1, 2009

Unit leader :

Graham Farr

Lecturer(s) :

Clayton

- Graham Farr

Malaysia

- Loke Kar Seng

Tutors(s) :

Clayton

- Robyn McNamara, Andre Oboler, Brendon Taylor

Introduction

Welcome to FIT1015 Computer Science. This is a 6 credit point unit and is a core unit in the Computer Science major in the Bachelor of Science. The unit is designed to develop the student's understanding on how to develop and use the basic data structures and algorithms, and also to explore how simple programs that use these basic components are actually executed by the computer.

Unit synopsis

FIT1015 Computer Science 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; introductory topics from software engineering; and translation from simple high level programs to assembler.

Learning outcomes

At the completion of this unit, students will have knowledge 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 computational 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.
- Gain a deeper understanding of basic object-oriented (OO) concepts, and understand more advanced ones such as inheritance, polymorphism, information hiding and encapsulation.
- Understand the design principles for building an object-oriented program, such as identify classes, and determine how and when to use inheritance.
- Understand the differences between the four major programming language paradigms: imperative, object oriented, functional and logic.
- Understand the software development life cycle. Analyse the advantages and disadvantages of different life cycle models.
- Understand the basic concepts in testing, including execution vs non-execution based testing, glass box and black box testing, correctness proofs, and test case selection. Analyse different testing approaches such as modular, static, dynamic, and formal.
- Understand the requirements for "good programming practice".
- Understand how numbers are represented on a computer.
- 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 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.

Students will have attitudes that make them:

- 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.

Students will have the practical skills to be able to:

- Create their own data-structures. Design and implement Java programs using a variety of data structures and algorithms.
- Implement an object-oriented program consisting of many interacting classes requiring not only basic but also advanced 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.
- Use the processor simulator for executing some of the simple iterative programs learned in this subject.
- Determine the time and space requirements of simple algorithms and data structures.

Students will also be able to:

- Document a program correctly.
- Produce appropriate documentation for designing and testing a program.
- Explain how parts of a program work.

Workload

For on campus students, workload commitments are:

- three one-hour lectures,
- one one-hour tutorial
- one 1 and 1/2 hour computer lab prac (requiring advance preparation) followed by an extra 1 and 1/2 hour for prac marking
- a minimum of 6 hours of personal study per week in order to satisfy the reading and assignment expectations.
- You will need to allocate up to 4 hours per week, for use of a computer, including time for newsgroups/discussion groups.

Unit relationships

Prerequisites

Before attempting this unit you must have satisfactorily completed:

- FIT1001 Computer Systems or equivalent
- FIT1002 Computer Programming or equivalent

Students beginning FIT1015 Computer Science 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 asimple problem.
- Use software development tools such as compilers, debuggers, editors. In particular, design, implement, compile, debug and execute a Java program containing selection, repetition, simple classes and one dimensional arrays.
- Perform decimal to binary and hexadecimal conversions, represent integers, signed fractional, floating point and character data.
- Perform basic boolean algebra and digital logic, form logic expressions, build truth tables, and understand logic gates.
- Follow the main steps in the fetch-decode-execute cycle.

Relationships

FIT1015 is a core unit in the Bachelor of Computer Science and in the Bachelor of Software Engineering.

FIT1015 is a prerequisite for:

- FIT2004 Algorithms and Data Structures
- FIT2008 Net-centric Computing
- FIT2025 Software Engineering Practice

You may not study FIT1015 and CSE1303, CSC1030, FIT1007, FIT1015 in your degree.

Continuous improvement

Monash is committed to 'Excellence in education' (Monash Directions 2025 - http://www.monash.edu.au/about/monash-directions/directions.html) and strives for the highest possible quality in teaching and learning.

To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. The University's Unit Evaluation policy (http://www.policy.monash.edu/policy-bank/academic/education/quality/unit-evaluation-policy.html) requires that every unit offered is evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to "have their say". The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

Faculties have the option of administering the Unit Evaluation survey online through the my.monash portal or in class. Lecturers will inform students of the method being used for this unit towards the end of the semester.

Student Evaluations

If you wish to view how previous students rated this unit, please go to
http://www.adm.monash.edu.au/cheq/evaluations/unit-evaluations/

Improvements to this unit

Two main changes have been made to the unit. The first one relates to the pracs and, in particular, to the fact that the first two pracs (held in weeks 2 and 3) will not be performed in pairs. The reason for this decision is the difficulties found during last semester in determining the pairs during the first weeks due to organisational (students changing their enrollments or their attendance to a given prac) and social issues (students who knew very few other students in the prac).

The second change relates to a change in the Learning Management System: we will use Moodle rather than the old Blackboard Vista. This change comes as a result of the number of student complaints regarding the problems in using Blackboard. As a result, we would like to ask students for feedback regarding the ease of use of Moodle and the adequacy of its capabilities.

Unit staff - contact details

Unit leader

Associate Professor Graham Farr
Associate Professor
Phone +61 3 990 55201
Fax +61 3 990 55146
Contact hours : Tuesdays 3-4pm, Wednesdays 4-5pm

Lecturer(s) :

Mr Kar Loke
Contact hours : Mondays 10am-12pm, Tuesdays 10am-12pm, Wednesdays 11am-12pm

Associate Professor Graham Farr
Associate Professor
Phone +61 3 990 55201
Fax +61 3 990 55146
Contact hours : Tuesdays 3-4pm, Wednesdays 4-5pm

Tutor(s) :

Robyn McNamara, Andre Oboler, Brendon Taylor

Additional communication information

The preferred communication method for questions regarding the unit's material and/or organisation is through the on-line discussion forum (that way, everyone can benefit from it). For more in-depth help, students can either talk to the lecturer during consultation hours, or through the "Help Room" sessions.

If a student has a personal matter that they wish to notify the lecturer about then they should email cl-fit1008-admin@infotech.monash.edu.au.

Notices related to the unit during the semester will be placed on the News of the Unit's Website.Please, check this regularly. Failure to read the Notices newsgroup is not regarded as grounds for special consideration.

IMPORTANT: forany e-mail contact regarding this unit, please make sure your subject line starts with "FIT1015:". Otherwise, the e-mail might go undetected.

Teaching and learning method

The main teaching mechanisms are through the material covered in lectures, and the questions and discussion promoted during tutorials. Learning is also expected to occur through discussions with the prac partner, interaction with the demonstrator, participation in the on-line discussion forum for the unit, and through the reading of the book chapters recomended for each topic in the lecture slides.

Timetable information

For information on timetabling for on-campus classes at Sunway Campus please refer to
http://www.monash.edu.my/timetables/infotech.html

Tutorial allocation

For tutorial and prac allocation please register using Allocate+

Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should

discuss it with your lecturer or a Community Service counsellor as soon as possible.

The main source of feedback for the unit is the feedback that you will get from the demonstrators at the pracs. During the first 1 and 1/2 hours the demonstrator will answer any questions you have regarding the prac and will advise you on the quality of your current solution. During the last 1 and 1/2 hours, the demonstrator will tell you the mark you have achieved in every question and the reasons for that mark.

Each demonstrator carries a marking guide given by the lecturer with clear specifications regarding how to mark each question. You can ask the demonstrator to show you the guide. However, note that any decisions made by the demonstrator at the prac are final. If you have a (reasonable) problem, you must take it up with the lecturer.

Unit Schedule

Week	Topic	Key dates
1	Revision FIT1002	
2	Sorting & Array Data Structures	
3	Linked Data Structures	
4	Object Oriented Basics	
5	Advanced OO	
6	Testing/Debugging	22nd August Mid-Semester Test
Mid semester break		
7	Recursive Sorts & Trees	
8	Trees	
9	Architecture, number representation	
10	MIPS	
11	Translating to assembler	
12	Function call/return	
13	Revision	

Unit Resources

Prescribed text(s) and readings

There are no required texts for this subject. The lecture slides are designed to be self sufficient. However, we strongly encourage students to read the appropriate parts of the recommended texts, which complement the slides.

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

(1) *Data Structures and Algorithms in Java*. Second Edition. 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.

(2) *Absolute Java*.Second Edition. Walter Savitch. Addison Wesley. This book also contains some data structures and algorithms, but it uses them to illustrate the use of Java. It is useful for students who have questions about the Java language.

(3) *Algorithms in Java*. Third Edition. Robert Sedgewick. Parts 1-4. This book is a more in-depth book. 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 Algorithms in Java.* Adam Drozdek, Brooks/Cole. More advanced.

Required software and/or hardware

Eclipse Platform. This is the recomended platform (although BlueJ is also allowed). It can be downloaded from http://www.eclipse.org/downloads/

BlueJ, Version 2.1.2 Programming Development Environment. It can be downloaded from http://www.bluej.org

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

The MIPS R2000 simulator **SPIM S20**. This, and all the other above, are included as part of the Standard Operating Environment used in Faculty computer Labs.

Equipment and consumables required or provided

Students 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 **4** 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:

found at the FIT1015 site, including:

- lecture slides,
- code for the lectures in Java class format,
- weekly tutorial exercises,
- weekly assignment specifications,
- weekly tutorial solutions (available *after* the tutorial), and
- supplementary material.

Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to http://www.lib.monash.edu.au.

The Educational Library and Media Resources (LMR) is also a very resourceful place to visit at http://www.education.monash.edu.au/library/

Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (http://moodle.monash.edu.au) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

You can access MUSO and Blackboard via the portal: http://my.monash.edu.au

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: http://www.monash.edu.au/muso/support/students/downloadables-student.html

You can contact the MUSO Support by phone : (+61 3) 9903 1268

For further contact information including operational hours, please visit:
http://www.monash.edu.au/muso/support/students/contact.html

Further information can be obtained from the MUSO support site:
http://www.monash.edu.au/muso/support/index.html

Assessment

Unit assessment policy

To pass this unit you must:

- attend at least 7 out of the 11 pracs;
- attend at least 7 out of the 11 tutorials;
- score 50% or better in pracs
- score 50% or better in the exam, and
- score at least 50% overall.

If these four hurdles are met, your score for the unit will be calculated by:

0.7(Total Exam Mark) + 0.2*(Total Prac Mark) + 0.1*(Total Test mark)*

Otherwise, the maximum score is 44N

Assignment tasks

- ## Assignment Task

  Title : Mid Semester Test (1 hour)

  Description :

  This test will evaluate your understanding of the material provided during the first five 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.

  Weighting : 10%

  Criteria for assessment :

  See previous paragraph (Description).

  Due date : 9 April

- ## Assignment Task

  Title : Pracs (1 and 1/2 hours each)

  Description :

  Each week you will need to complete a prac assignment together with another student. Prac assignments are long and are designed to take a significant part of your 6 ``home study hours'' (usually, up to 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 and 1/2 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 the1 and 1/2 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 in 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 : Each prac will be marked during the hour-and-a-half immediately after the first hour-and-a-half of that prac session. You must remain in the prac session until your prac is marked.

  Remarks (optional - leave blank for none) :

  There are two hurdles associated to the pracs. First, you must attend at least 7 out of the 11 pracs. Second, you must score at least 50% of the prac mark.

  There is one hurdle associated with tutorials. You must attend at least 7 out of the 11 tutorials.

  A student who does not meet all these hurdles can get a maximum of 44 N for the unit.

Examinations

- **Examination 1**

  Weighting : 70%

  Length : 3 hours

  Type (open/closed book) : Closed book

  Remarks (optional - leave blank for none) :

  There is a hurdle associated with the exam mark: you must score at least 50% of the exam mark. Furthermore, you must score at least 50% overall (i.e., for the mid semester test, pracs and exam).

Assignment submission

The solution provided by each student (or pair of students) for the weekly prac will be reviewed and marked by a Lab demonstrator at the weekly prac during the last 1 and 1/2 hours of the prac. This is done in front of the students and at the student's computer. Demonstrators are obliged to tell students the mark obtained in each question and the reasons for giving such mark.

The code marked during the prac must be submitted by electronic submission to the FIT1015 MUSO site right after being marked. Failure to do this will result in the prac being given 0 marks.

Assignment coversheets

When submmitting your assignment, make sure you include an assignment coversheet. An electronic version of the coversheet can be found at the unit's site.

University and Faculty policy on assessment

Due dates and extensions

The due dates for the submission of assignments are given in the previous section. 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 seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

If for any valid reason you cannot attend your usual pracand would like to attend a different prac, you need to emailcl-fit1008-admin@infotech.monash.edu.au and ask for permission. In doingthis you must provide:

- Your name:
- Your ID number:
- Regular Prac: (time and room)
- The reason you cannot attend your usual prac.
- The details (time and room) of the prac you would like to attend,
- The name of your prac partner, and
- Evidence that thepartner knows of the change (an email message is acceptable) and will also be able attend thealternative prac.

Late assignment

If you miss a prac or tutorial class for any reason you *must* do the following to obtain an exemption for the missed class:

- Submit a special consideration form no more than one week after you return to University. These forms are available from and should be handed in to the General office (Clayton) in building 63.
- Attach any documentary evidence, for example, medical certificate covering the date of your missed class, letter of explanation, police report or plane boarding pass.

Failure to do the above will result in you being marked absent for the class and receiving zero marks. Exemptions will not be granted automatically, and will be considered on a case by case basis.

Return dates

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

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at http://www.policy.monash.edu/policy-bank/academic/education/assessment/

Since the pracs are marked weekly during the lab session, feedback will be provided immediately during the marking.

The mid semester test will be returned within two weeks of the test date.

Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with the University Plagiarism policy and procedure (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html) which applies to students detected plagiarising.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

Register of counselling about plagiarism

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

Non-discriminatory language

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

Students with disabilities

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- Disabilities Liaison Unit

Deferred assessment and special consideration

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at http://www.monash.edu.au/exams/special-consideration.html. Contact the Faculty's Student Services staff at your campus for further information and advice.