



MONASH University

FIT2034
Computer programming 2

Unit guide

Semester 1, 2009

Last updated : 12 Feb 2009

Table of Contents

<u>FIT2034 Computer programming 2 - Semester 1, 2009</u>	1
<u>Unit leader</u> :.....	1
<u>Lecturer(s)</u> :.....	1
<u>Caulfield</u>	1
<u>Malaysia</u>	1
<u>Tutors(s)</u> :.....	1
<u>Caulfield</u>	1
<u>Introduction</u>	1
<u>Unit synopsis</u>	1
<u>Learning outcomes</u>	2
<u>Workload</u>	2
<u>Unit relationships</u>	3
<u>Prerequisites</u>	3
<u>Relationships</u>	3
<u>Continuous improvement</u>	3
<u>Student Evaluations</u>	4
<u>Improvements to this unit</u>	4
<u>Unit staff - contact details</u>	4
<u>Unit leader</u>	4
<u>Lecturer(s)</u> :.....	4
<u>Tutor(s)</u> :.....	4
<u>Teaching and learning method</u>	4
<u>Timetable information</u>	5
<u>Tutorial allocation</u>	5
<u>Communication, participation and feedback</u>	5
<u>Unit Schedule</u>	5
<u>Unit Resources</u>	6
<u>Prescribed text(s) and readings</u>	6
<u>Recommended text(s) and readings</u>	6
<u>Required software and/or hardware</u>	6
<u>Equipment and consumables required or provided</u>	7
<u>Study resources</u>	7
<u>Library access</u>	7
<u>Monash University Studies Online (MUSO)</u>	7
<u>Assessment</u>	8
<u>Unit assessment policy</u>	8
<u>Assignment tasks</u>	8
<u>Examinations</u>	8
<u>Assignment submission</u>	9
<u>Assignment coversheets</u>	9
<u>University and Faculty policy on assessment</u>	9
<u>Due dates and extensions</u>	9
<u>Late assignment</u>	9
<u>Return dates</u>	9
<u>Plagiarism, cheating and collusion</u>	10
<u>Register of counselling about plagiarism</u>	10
<u>Non-discriminatory language</u>	10
<u>Students with disabilities</u>	11
<u>Deferred assessment and special consideration</u>	11

FIT2034 Computer programming 2 - Semester 1, 2009

Unit leader :

Chris Ling

Lecturer(s) :

Caulfield

- Chris Ling

Malaysia

- Mylini Munusamy

Tutors(s) :

Caulfield

- Pari Delir Haghighi
- Eddie Leung
- Chris Ling

Introduction

Welcome to FIT2034 Computer Programming 2 for Semester 1, 2009. This 6 point unit is a core unit in the following majors of the Bachelor of Information Technology and Systems (BITS) degree:

- *Applications Development and Networks*
- *Multimedia Games Development*
- *Net-centric Computing*
- *System Development*
- *Security*

The unit has been introduced to supplement your first programming unit FIT1002 to provide you with an understanding of more advanced concepts in object-oriented programming.

Unit synopsis

This unit introduces more advanced object-oriented programming topics and techniques than its prerequisite, and gives students a deeper understanding of programming and data structures and more practical skills in designing, building and testing computer programs.

Learning outcomes

At the completion of this unit, students will have an understanding of:

- K1. Object-oriented concepts such as inheritance, polymorphism, and abstract classes.
- K2. The implementation in Java of object-oriented concepts such as multiple inheritance.
- K3. How to test a program consisting of many interacting classes.
- K4. The collection classes in the Java API.
- K5. Design principles for building an object-oriented program.
- K6. Problem-solving techniques for debugging an object-oriented program.
- K7. The concept of recursion in a computer program.
- K8. Dynamic data structures.

At the completion of this unit, students will have attitudes that will allow them to:

- A1. Write programs that conform to programming standards
- A2. Use good design principles when constructing systems
- A3. Take a patient and thorough approach to testing
- A4. Acknowledge any assistance they have received in writing a program
- A5. Search for supplementary unit-related information in appropriate places when necessary

At the completion of this unit, students will be able to:

- P1. Design an object-oriented program consisting of many interacting classes with association, generalization and aggregation relationships.
- P2. Construct a test harness for testing a multiple class object-oriented program.
- P3. Write code to implement a multiple class object-oriented design in Java including association, generalization and aggregation relationships.
- P4. Debug and modify an existing program consisting of many interacting classes.
- P5. Use the Java API classes as part of their programs.
- P6. Use the Java collection classes to store and retrieve data appropriately.
- P7. Use recursion to solve new problems
- P8. Read data from keyboard and files, and write data to screen and files.
- P9. Use exception-handling techniques in programs.
- P10. Use event-handling techniques in GUI programs.
- P11. Use UML to design an object-oriented program.
- R1. Document a program correctly
- R2. Produce appropriate documentation for designing and testing a program
- R3. Explain how parts of a program work

Workload

For on campus students, workload commitments are:

- two-hour lecture and
- two-hour tutorial (or laboratory)
- 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

Before attempting this unit you must have satisfactorily completed either CSE1202 or GCO1811 or CPE1001 or FIT1002 or equivalent.

Students beginning Computer Programming 2 are assumed to be able to:

- Design an object-oriented program using class diagrams
- Implement a design in Java using multiple interacting classes and techniques such as sequence, selection, repetition, parameter passing, scoping, recursion
- Debug an existing small program
- Test a small object-oriented program thoroughly
- Read existing Java code (at a basic level)
- Integrate existing classes into their programs
- Produce basic program- and user-documentation to taught standards

Relationships

FIT2034 is a core unit in the majors of the Bachelor of Information Technology and Systems. It may be taken as an elective in other programs where you have satisfied the prerequisites and where course rules permit.

It is a prerequisite for:

- FIT2009 Data structures and algorithms
- FIT2024 Software engineering practice
- FIT3061 Handheld applications and operating systems
- FIT3011 Enterprise programming

Before attempting this unit you must have satisfactorily completed either CSE1202, GCO1811, CPE1001, FIT1002 or equivalent.

You may not study this unit if you have studied or intend to study any of the following units in your degree: CFR1124, CFR2128, CFR3112, GCO1812, GCO3821, SFT1102, CSE1203, CSE2305, CPE1004, FIT1008, FIT1007, FIT1015.

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.monash.edu.au/unit-evaluation-reports/>

Improvements to this unit

1. To improve the sequencing of certain topics.
2. To change the assignment weightage.
3. To focus on programming skills rather than the implementation of prototype features in all practical assignments. To align the assessment requirements with student's understanding of programming concepts rather than prototype features. The alignment will be done based on weekly learning objectives.
4. To propose a change of prescribed text to:
"Developing Java Software" 3rd Edition, by Russel Winder and Graham Roberts, ISBN 0-470-09025-1

Unit staff - contact details

Unit leader

Dr Chris Ling

Senior Lecturer

Phone +61 3 990 32808

Fax +61 3 990 31077

Lecturer(s) :

Dr Chris Ling

Senior Lecturer

Phone +61 3 990 32808

Fax +61 3 990 31077

Ms Mylini Munusamy

Tutor(s) :

Dr Chris Ling

Senior Lecturer

Phone +61 3 990 32808

Fax +61 3 990 31077

Mr Eddie Leung

Mrs Pari Delir Haghighi

Teaching and learning method

The approach to teaching and learning include a weekly two-hour lecture and a two-hour (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

Caulfield students should check the timetable using Monash Timetabling System:
<http://mutts.monash.edu.au/MUTTS/>

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

Tutorial allocation

On-campus students should register for tutorials/laboratories 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.

Unit Schedule

Week	Topic	Key dates
1	Revising Java Concepts	
2	Designing objects	
3	Association/Aggregation relationships	Week 2 Exercise Due
4	Inheritance and Polymorphism	Week 3 Exercise Due
5	Interfaces and Abstract Classes	Week 4 Exercise Due
6	GUI and Event Handling	Week 5 Exercise Due
Mid semester break		
7	Recursion	Week 6 Exercise Due
8	Exceptions	Week 7 Exercise Due
9	File Input/Output	
10	Utility Classes	Week 9 Exercise Due
11	Dynamic Data Structures, ADTs and Collection Classes	
12	Testing and Debugging	Programming Assignment Due on Fri 29 May
13	Revision	

Unit Resources

Prescribed text(s) and readings

Prescribed Text:

- Russel Winder and Graham Roberts (2006). Developing Java Software. 3rd Edition (ISBN 0-470-09025-1)

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

- Horstmann, C. (2005) Big Java, 3rd edition, Wiley & Sons. (ISBN: 0-470-10554-2)
- Savitch, W. (2006) Absolute Java, 2nd edition, Addison Wesley. ISBN: 0-321-33024-2
- Knox-Grant, R. (2005) Java 2 Beyond the Buttons, 2nd Ed., Greenoil, ISBN: 0-620-33395-2
- Robertson, L. (2003) Simple program design : a step by step approach, 4th edition, Thomson. ISBN: 0-17-010704-3
- Nino, J. (2005) An Introduction to Programming and Object Oriented Design Using Java Version 5.0, Wiley. ISBN: 0-471-71227-2

Required software and/or hardware

Prescribed Software

You must have the Java 2 SE SDK version 1.5.0 (also called Java 5) or later installed on your computer. It can also be downloaded from the internet by going to <http://java.sun.com/javase/downloads/index.jsp>

Other Useful Software

BlueJ

Some locations will be using BlueJ. This also works on Macintosh computers. It can also be downloaded from <http://www.bluej.org/download/download.html>.

JCreator LE

This is an IDE which provides many useful compilation features. It only works on Windows operating systems. The smallish download can be obtained from <http://www.jcreator.com/download.htm>. Be sure to select the LE version 3.5 file, which is free (unless you want to pay for the more comprehensive version).

TortoiseSVN

For experienced programmers who want a source-code management system that runs on Windows, TortoiseSVN is strongly recommended. This open-source software can be downloaded from: <http://tortoisesvn.sourceforge.net/downloads>. (Most users should download the very first file, the 32-bit msi file). There are also language-packs for languages other than english. On campus classes might demonstrate this tool at some point in the semester.

Equipment and consumables required or provided

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 **10** 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:

- This *Unit Information Guide* outlining the administrative information for the unit
- The FIT2034 web site on Moodle, where lecture slides, weekly tutorial requirements, assignment specifications, sample solutions and supplementary material will be posted.
- Discussion Groups.

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, a student must obtain:

- 40% or more in the unit's examination; and
- 40% or more in the unit's non-examination assessment (assignments total); 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 assessment, then a mark of no greater than 44-N will be recorded for the unit.

Assignment tasks

• Assignment Task

Title : Programming Assignment

Description :

Non-trivial programming assignment

Weighting : 20%

Criteria for assessment :

The specification and marking criteria will be distributed in week 3.

Due date : Friday 29 May 2009

• Assignment Task

Title : Lab Exercises

Description :

Weekly exercises in the lab. The total number of exercises to be completed is 7. See the Unit Schedule table the weekly exercises that will be assessed.

Weighting : 20%

Criteria for assessment :

Assessment criteria will be distributed in that week.

Due date : Exercise distributed in the week will be due at the beginning of the following week's tute class

Examinations

• Examination 1

Weighting : 60%

Length : 3 hours

Type (open/closed book) : closed book

Assignment submission

Assignments will be submitted electronically using MUSO or WebFace (gippsland only). The due date is the date/time by which the submission must be received/the date by which the the submission is to be posted.

Assignment coversheets

The work submitted must be your own work. Just as a written assignment requires you to sign a statement that the work is your own work, electronically submitted assignments must also contain a similar declaration. The cover sheet for the Faculty of Information Technology, can be found at <http://www.infotech.monash.edu.au/resources/student/assignments/> . The text of the section titled "Student's Statement" should be typed into a file named **declaration.txt**, to be submitted with your other files.

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.

Requests for extensions must be made to the unit lecturer at your campus at least two days before the due date. You will be asked to forward original medical certificates in cases of illness, and may be asked to provide other forms of documentation where necessary. A copy of the email or other written communication of an extension must be attached to the assignment submission.

Late assignment

Assignments received after the due date will be subject to a penalty of a drop in grade for each 5 day period. Assignments received later than one week after the due date will not normally be accepted.

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/>

We will aim to have assignment results made available to you within two weeks after assignment receipt. Even if there is a delay in returning your assignment result, the sample solution will be made available within two weeks after the due 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 Student Rights and Responsibilities (<http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html>) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

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.