



MONASH University
Information Technology

FIT2034
Computer programming 2

Unit Guide

Semester 2, 2010

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: 06 Jul 2010

Table of Contents

<u>FIT2034 Computer programming 2 - Semester 2, 2010</u>	1
<u>Chief Examiner:</u>	1
<u>Lecturer(s) / Leader(s):</u>	1
<u>Caulfield</u>	1
<u>Gippsland</u>	1
<u>South Africa</u>	1
<u>Malaysia</u>	1
<u>Additional communication information:</u>	1
<u>Introduction</u>	3
<u>Unit synopsis</u>	3
<u>Learning outcomes</u>	3
<u>Contact hours</u>	3
<u>Workload</u>	3
<u>Unit relationships</u>	4
<u>Prerequisites</u>	4
<u>Prohibitions</u>	4
<u>Teaching and learning method</u>	5
<u>Teaching approach</u>	5
<u>Timetable information</u>	5
<u>Tutorial allocation</u>	5
<u>Off-Campus Learning or flexible delivery</u>	5
<u>Unit Schedule</u>	5
<u>Improvements to this unit</u>	6
<u>Unit Resources</u>	7
<u>Prescribed text(s) and readings</u>	7
<u>Recommended text(s) and readings</u>	7
<u>Required software and/or hardware</u>	7
<u>Equipment and consumables required or provided</u>	8
<u>Study resources</u>	8
<u>Assessment</u>	9
<u>Overview</u>	9
<u>Faculty assessment policy</u>	9
<u>Assignment tasks</u>	9
<u>Examination</u>	12
<u>Due dates and extensions</u>	12
<u>Late assignment</u>	12
<u>Return dates</u>	12
<u>Feedback</u>	13
<u>Appendix</u>	14

FIT2034 Computer programming 2 - Semester 2, 2010

Chief Examiner:

Dr Chris Ling

Senior Lecturer
Phone: +61 3 990 32808
Fax: +61 3 990 31077

Lecturer(s) / Leader(s):

Caulfield

Dr Chris Ling

Senior Lecturer
Phone: +61 3 990 32808
Fax: +61 3 990 31077

Gippsland

Mr Shane Moore

Lecturer
Phone: +61 3 990 26716

South Africa

Ms Sheelagh Walton

Lecturer
Phone: +27 11 950 4034
Fax: +27 11 950 4033

Malaysia

Ms Mylini Munusamy

Additional communication information:

Unless you have personal enquiries (see below) all communication related to the content of the unit must be via the online Discussion Forums. If you do send the lecturer an email that relates to the content of the unit it might not be answered, or you may be told to ask to the forum.

Personal enquiries include seeking advice regarding assignment extensions (where warranted by circumstances recognised for special consideration requests), or the need to discuss your personal progress. You are certainly not asked to put anything of a personal nature into forum postings. Personal matters can also be dealt with by telephone.

On-campus students, and off-campus students who live or work near a campus, may also visit their lecturer at their office.

Note: The staff may contact you during the semester, by sending an email to your **@student.monash.edu** address. You are therefore expected to either check that email regularly (at least twice a week), or have it redirect mail to an address which you are going to check regularly. Also, any email purporting to be from a student which does not come from your Monash email address are allowed to be ignored by the staff member, as sending replies to any other addresses could be a violation of the Privacy provisions of legislation.

Introduction

Welcome to FIT2034 Computer Programming 2. 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

Following on from [FIT1002](#), this unit introduces more advanced object-oriented programming topics than its prerequisite, such as inheritance and polymorphism. It gives students a deeper understanding of programming and data structures by introducing recursion and dynamic data structures. It also gives more practical skills in designing, building and testing larger computer programs, including ones having graphical user interfaces, and utilising file I/O. Modern software tools to support programming activities of testing and group-based development are also demonstrated.

Learning outcomes

At the completion of this unit students will:

- demonstrate an understanding of advanced object-oriented concepts such as inheritance, polymorphism, and abstract classes and interfaces as provided for in Java;
- be able to create programs that provide a graphical user interface and use event handling;
- be able to write programs involving abstract and dynamic data structures, and implement algorithms for searching, insertion and deletion;
- be able to use the collection classes in the Java API;
- be able to implement algorithms that utilise recursion;
- have an understanding of design principles for building a multiple-class object-oriented program;
- be able to implement exception handling techniques;
- be able to use files for persistent storage of data;
- be able to construct test harnesses for multiple-class programs;
- demonstrate an understanding of the range and purpose of modern tools to support the process of programming complex software systems.

Contact hours

2 hrs lectures/wk, 2 hrs laboratories/wk

Workload

For on campus students, workload commitments are:

- two-hour lecture;
- two-hour 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 discussion groups.

Off-campus students generally do not attend lecture and tutorial sessions, but should plan to spend equivalent time working through the relevant resources and participating in discussion groups each week. Being a 6-point unit, an average of 12 hours per week should be spent on this unit.

Unit relationships

Prerequisites

FIT1002 or CPE1001 or CSE1202 or GCO1811 or equivalent

Prohibitions

CPE1004, CSE1203, CSE2305, GCO1812, FIT1007

Teaching and learning method

Teaching approach

The curriculum is defined by what is covered in the Study Guide modules for each week. All students are expected to work through the study guide modules in preparation for that week's classes.

On campus students will have two classes per week: a lecture/workshop session and a practical lab session. The purpose of the first is to discuss the concepts of that week's curriculum. The purpose of the second is to enable you to apply the concepts by working on problems on a computer.

Assignments are designed to be attempted **after** you have completed all required readings and practical exercises. They are also likely to be another source of learning, although their primary purpose is for staff to assess how well you have progressed in meeting the learning objectives of the unit.

Discussion forums are provided as a place where you may ask questions about the content of the unit. You should also use these to clarify the work required in your assignments. They are checked at least twice per week, and sometimes even more often than that.

Students should spend approximately 8 hours per week outside of class for personal study every week, including time for programming on a computer and reading the discussion forums.

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.its.monash.edu.au/>

Off-Campus Learning or flexible delivery

Off-campus students should refer to the study guide modules as their primary source of direction for studying in this unit. The study guide modules tell you what readings from the textbook to read and convey other thoughts of the teaching staff.

All off-campus students are expected to keep up to the schedule as outlined in this document, and attempt the ungraded practical tasks (in addition to the few weeks which are graded). You can share/discuss your attempts online, and if you have queries about how to attempt them, post these to the discussion forum.

Unit Schedule

Week	Date*	Topic	Study guide	References/Readings	Key dates
1	19/07/10	Revising Java Concepts	1	Reges ch 1-5	No lab in Week 1
2	26/07/10	Object Orientation	2	Reges ch 8	Lab classes start this week

3	02/08/10	Association and Aggregation Relationships	3	Reges ch 7 and 10.1	Week 2 Exercise Due
4	09/08/10	Inheritance and Polymorphism	4	Reges ch 9	
5	16/08/10	Interfaces and Abstract Classes	5	Reges ch 9	Week 4 Exercise Due
6	23/08/10	File I/O and Exceptions	6	Reges ch 6	
7	30/08/10	Abstract Data Types and the Java Collection Classes	7	Reges ch 11	
8	06/09/10	Recursion	8	Reges ch 12	Week 7 Exercise Due
9	13/09/10	Algorithms for Searching and Sorting	9	Reges ch 13	
10	20/09/10	Event Handling and Graphical User Interfaces	10	Reges ch 14	
Mid semester break					
11	04/10/10	Testing and Debugging	11	Binder ch 1 & 3 (online)	Week 10 Exercise Due
12	11/10/10	Program Design	12		Major Assignment Due
13	18/10/10	Revision	All		

*Please note that these dates may only apply to Australian campuses of Monash University. Off-shore students need to check the dates with their unit leader.

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.

Unit Resources

Prescribed text(s) and readings

Prescribed Text:

- Reges, S. (2011) "Building Java Programs : A Back to Basics Approach", Second Edition, Addison Wesley (ISBN 0-136-09181-4)

However, if you already have access to the First Edition, that will be acceptable.

Recommended text(s) and readings

Arnold, K., et al (2006), "The Java Programming Language", 4th edition, Sun Microsystems/Addison-Wesley. (ISBN: 0-321-34980-6)

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 4.5 file, which is free (unless you want to pay for the more comprehensive version).

Eclipse

Enthusiastic/Advanced students may prefer to use Eclipse as their IDE. This can be downloaded from <http://www.eclipse.org/>

TortoiseSVN

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. This tool is explained in week 12 of 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
- Weekly Study Guides modules
- Weekly slides and sample programs used during lectures
- Weekly practical tasks and solutions
- Assignment specification
- Additional electronic resources
- Discussion forums
- The FIT2034 web site on Moodle, where most of the above resources can be located

Assessment

Overview

Examination (3 hours): 60%; In-semester assessment: 40%

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 50% then a mark of no greater than 49-N will be recorded for the unit.

Assignments have a due date and a cutoff date. The due date is that time by which you are required to submit the assignment to receive the grade that it deserves. Submitting after the due date but before the cutoff date means that there will be a penalty applied to your result (typically, a drop in grade by one level). Submissions received after the cutoff date will receive 0 marks. The cutoff date is usually 1 week later than the due date. If you are given an extension, the extension applies to both the due date and the cutoff date.

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 submission and preparation requirements will be detailed in each assignment specification. 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>.

• Assignment task 1

Title:

Week 2 Exercise

Description:

The individual exercise is based on Week 2 topic. The specification will be released in the same week.

Weighting:

5%

Criteria for assessment:

The criteria used to assess submissions are:

1. The program must compile and run correctly.
2. The program must meet the problem specification.
3. Java code should be readable and maintainable.
4. The program should be documented.
5. All algorithms should follow the style presented in lectures and be correct.

On-campus students must demo their programs to the tutor in the following week's laboratory class as part of the assessment. During the demo, they must be able to answer questions asked about their programs.

Due date:

Week 3

• **Assignment task 2**

Title:

Week 4 Exercise

Description:

This individual exercise is based on Week 4 topic. The specification will be released in the same week.

Weighting:

5%

Criteria for assessment:

The criteria used to assess submissions are:

1. The program must compile and run correctly.
2. The program must meet the problem specification.
3. Java code should be readable and maintainable.
4. The program should be documented.
5. All algorithms should follow the style presented in lectures and be correct.

On-campus students must demo their programs to the tutor in the following week's laboratory class as part of the assessment. During the demo, they must be able to answer questions asked about their programs.

Due date:

Week 5

• **Assignment task 3**

Title:

Week 7 Exercise

Description:

The individual exercise is based on Week 7 topic. The specification will be released in the same week.

Weighting:

5%

Criteria for assessment:

The criteria used to assess submissions are:

1. The program must compile and run correctly.
2. The program must meet the problem specification.
3. Java code should be readable and maintainable.
4. The program should be documented.

5. All algorithms should follow the style presented in lectures and be correct.

On-campus students must demo their programs to the tutor in the following week's laboratory class as part of the assessment. During the demo, they must be able to answer questions asked about their programs.

Due date:

Week 8

• **Assignment task 4**

Title:

Week 10 Exercise

Description:

The individual exercise is based on Week 10 topic. The specification will be released in the same week.

Weighting:

5%

Criteria for assessment:

The criteria used to assess submissions are:

1. The program must compile and run correctly.
2. The program must meet the problem specification.
3. Java code should be readable and maintainable.
4. The program should be documented.
5. All algorithms should follow the style presented in lectures and be correct.

On-campus students must demo their programs to the tutor in the following week's laboratory class as part of the assessment. During the demo, they must be able to answer questions asked about their programs.

Due date:

Week 11

• **Assignment task 5**

Title:

Major Assignment

Description:

Each student is required to develop a multiple class program which involves the use of concepts from Week 1 to Week 10.

Weighting:

20%

Criteria for assessment:

The general criteria used to assess submissions are:

1. The program must compile and run correctly.
2. The program must meet the problem specification.
3. Java code should be readable and maintainable.
4. The program should be documented.
5. All algorithms should follow the style presented in lectures and be correct.

A more detailed assessment guide will be released together with the specification.

On-campus students must be interviewed individually by their tutor and must demo his/her program to the tutor. During the interview, they must be able to answer questions asked

about their programs.

Due date:

Sun 17 Oct 11.55pm

Examination

•

Weighting:

60%

Length:

3 hours

Type (open/closed book):

closed book

Electronic devices allowed in the exam:

None

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

Assignments received after the due date will be subject to a penalty of a drop in grade from what it is worth. Assignments received later than one week after the due date will not normally be accepted.

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.

Feedback

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

Informal feedback on progress in labs/tutes

Graded assignments with comments

Solutions to tutes, labs and assignments

Responses to queries posted in forums

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