



MONASH University
Information Technology

FIT2034
Computer programming 2

Unit Guide

Semester 1, 2011

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: 27 Feb 2011

Table of Contents

<u>FIT2034 Computer programming 2 - Semester 1, 2011</u>	1
<u>Mode of Delivery</u>	1
<u>Contact Hours</u>	1
<u>Workload</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>Caulfield</u>	2
<u>Learning Objectives</u>	2
<u>Graduate Attributes</u>	2
<u>Assessment Summary</u>	2
<u>Teaching Approach</u>	3
<u>Feedback</u>	3
<u>Our feedback to You</u>	3
<u>Your feedback to Us</u>	3
<u>Previous Student Evaluations of this unit</u>	3
<u>Required Resources</u>	3
<u>Recommended Resources</u>	4
<u>Examination material or equipment</u>	4
<u>Unit Schedule</u>	4
<u>Assessment Policy</u>	5
<u>Assessment Tasks</u>	5
<u>Participation</u>	5
<u>Examinations</u>	8
<u>Examination 1</u>	8
<u>Assignment submission</u>	8
<u>Extensions and penalties</u>	8
<u>Returning assignments</u>	8
<u>Policies</u>	8
<u>Student services</u>	9

FIT2034 Computer programming 2 - Semester 1, 2011

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.

Mode of Delivery

Caulfield (Day)

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

Prohibitions

CPE1004, CSE1203, CSE2305, GCO1812, FIT1007

Prerequisites

[FIT1002](#) or CPE1001 or CSE1202 or GCO1811 or equivalent

Chief Examiner

Chris Ling

Campus Lecturer

Caulfield

Chris Ling

Learning Objectives

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.

Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:
 - a. engage in an internationalised world
 - b. exhibit cross-cultural competence
 - c. demonstrate ethical values

critical and creative scholars who:

- a. produce innovative solutions to problems
- b. apply research skills to a range of challenges
- c. communicate perceptively and effectively

Assessment Summary

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

Assessment Task	Value	Due Date
Week 2 Exercise	5%	Week 3
Week 4 Exercise	5%	Week 5
Week 7 Exercise	5%	Week 8
Week 9 Exercise	5%	Week 10

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

Feedback

Our feedback to You

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

- Informal feedback on progress in labs/tutes
- Graded assignments with comments
- Interviews
- Solutions to tutes, labs and assignments
- Other: Responses to queries posted in forums

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 SETU, Student Evaluation of Teacher and Unit. 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, and on student evaluations, see:

<http://www.monash.edu.au/about/monash-directions/directions.html>

<http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html>

Previous Student Evaluations of this unit

If you wish to view how previous students rated this unit, please go to

<https://emuapps.monash.edu.au/unitevaluations/index.jsp>

Required Resources

Prescribed Software

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

Recommended Resources

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.

Examination material or equipment

It is a close book exam. No material or equipment is permitted.

Unit Schedule

Week	Date*	Activities	Assessment
0	21/02/11	Register for tutorials and check out the unit website	No formal assessment or activities are undertaken in week 0
1	28/02/11	Revising Java Concepts	No lab in Week 1
2	07/03/11	Object Orientation	Lab classes start this week
3	14/03/11	Association and Aggregation Relationships	Week 2 Exercise Due
4	21/03/11	Inheritance and Polymorphism	
5	28/03/11	Interfaces and Abstract Classes	Week 4 Exercise Due
6	04/04/11	File I/O and Exceptions	
7	11/04/11	Abstract Data Types and the Java Collection Classes	
8	18/04/11	Recursion	Week 7 Exercise Due
Mid semester break			
9	02/05/11	Event Handling and Graphical User Interface	
10	09/05/11	Algorithms for Searching and Sorting	Week 9 Exercise Due

11	16/05/11	Testing and Debugging	
12	23/05/11	Program Design and Exam Revision	Major Assignment Due ON Friday 27 May 2011, 11.55pm
	30/05/11	SWOT VAC	No formal assessment is undertaken SWOT VAC

*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.

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

Assessment Tasks

Participation

• Assessment 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.
6. Program should be able to create objects and all methods of that object tested.

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

• **Assessment 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.
6. Program should be able to create objects and all methods of that object tested.

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

• **Assessment 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.
6. Program should be able to create objects and all methods of that object tested.

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

• **Assessment task 4**

Title:

Week 9 Exercise

Description:

The individual exercise is based on Week 9 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.
6. Program should be able to create objects and all methods of that object tested.

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 10

• **Assessment 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.
6. Program should be able to create objects and all methods of that object tested.

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:

Fri 27 May 2011, 11.55pm

Examinations

• Examination 1

Weighting:

60%

Length:

3 hours

Type (open/closed book):

closed book

Electronic devices allowed in the exam:

None

Assignment submission

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.

Extensions and penalties

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

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process:

<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>.

Returning assignments

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

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:

<http://policy.monash.edu.au/policy-bank/academic/education/index.html>

Key educational policies include:

- Plagiarism
(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html>)
- Assessment
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-p>)
- Special Consideration
(<http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h>)
- 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/key-dates/>);
- Orientation and Transition (<http://www.infotech.monash.edu.au/resources/student/orientation/>); and
- Academic and Administrative Complaints and Grievances Policy (<http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy>)

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 www.monash.edu.au/students. The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to <http://www.lib.monash.edu.au> or the library tab in my.monash portal for more information. 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://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html>;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

Study Resources

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

READING LIST

Prescribed Text

Reges, S. and M. Stepp (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

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