



MONASH University
Information Technology

FIT2034
Computer programming 2

Unit Guide

Semester 2, 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: 22 Aug 2011

Table of Contents

| | |
|---|-----------|
| <u>FIT2034 Computer programming 2 - Semester 2, 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> | 2 |
| <u>Campus Lecturer</u> | 2 |
| <u>Caulfield</u> | 2 |
| <u>Gippsland</u> | 2 |
| <u>South Africa</u> | 2 |
| <u>Sunway</u> | 2 |
| <u>Tutors</u> | 2 |
| <u>Caulfield</u> | 2 |
| | |
| <u>Academic Overview</u> | 3 |
| <u>Learning Objectives</u> | 3 |
| <u>Graduate Attributes</u> | 3 |
| <u>Assessment Summary</u> | 3 |
| <u>Teaching Approach</u> | 4 |
| <u>Feedback</u> | 4 |
| <u>Our feedback to You</u> | 4 |
| <u>Your feedback to Us</u> | 4 |
| <u>Previous Student Evaluations of this unit</u> | 4 |
| <u>Required Resources</u> | 4 |
| <u>Recommended Resources</u> | 5 |
| <u>Examination material or equipment</u> | 5 |
| | |
| <u>Unit Schedule</u> | 6 |
| | |
| <u>Assessment Requirements</u> | 7 |
| <u>Assessment Policy</u> | 7 |
| <u>Assessment Tasks</u> | 7 |
| <u>Participation</u> | 7 |
| <u>Examinations</u> | 10 |
| <u>Examination 1</u> | 10 |
| <u>Assignment submission</u> | 11 |
| <u>Extensions and penalties</u> | 11 |
| <u>Returning assignments</u> | 11 |
| | |
| <u>Other Information</u> | 12 |
| <u>Policies</u> | 12 |
| <u>Student services</u> | 12 |

FIT2034 Computer programming 2 - Semester 2, 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)
- Gippsland (Day)
- Gippsland (Off-campus)
- Sunway (Day)
- South Africa (Day)

Contact Hours

2 hrs lectures/wk, 2 hrs laboratories/wk

Workload

Students will be expected to spend a total of 12 hours per week during semester on this unit as follows:

For on-campus students:

Lectures: 2 hours per week

Tutorials/Lab Sessions: 2 hours per week per tutorial

and up to an additional 8 hours in some weeks for completing lab and project work, private study and revision.

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

Prohibitions

CPE1004, CSE1203, CSE2305, GCO1812, FIT1007

Prerequisites

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

Chief Examiner

Mr Shane Moore

Campus Lecturer

Caulfield

Dhananjay Thiruvady

Gippsland

Shane Moore

South Africa

Sheelagh Walton

Sunway

Tan Choon Ling

Tutors

Caulfield

Matthew Kairys

Academic Overview

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 |
|---|-------|----------|
| Topic 2 and 3 Practical: Simple Object Oriented program involving associations and aggregations | 5% | Week 4 |
| Topic 4: Inheritance and Polymorphism - Practical | 4% | Week 5 |
| Topic 6: File Input and Output - Practical | 4% | Week 7 |
| Topic 7: Data Structures and the Java Collection Classes - Practical | 4% | Week 8 |
| Topic 8: Recursion - Practical | 4% | Week 9 |
| Topic 9: Graphical User Interfaces - practical | 4% | Week 10 |

Academic Overview

| | | |
|------------------|-----|---------------------------|
| Major Assignment | 15% | Sunday 23 October 2011 |
| Examination 1 | 60% | To be advised |

Teaching Approach

Lecture and tutorials or problem classes

Lectures are used to present new programming language concepts, and to present example code that uses these concepts.

Practicals are used to give you hands-on experience at programming using the newly taught concepts.

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: Staff responses to queries posted in discussion-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 SE 6 Software Development Kit (called the JDK) installed on your computer. This software can be downloaded for free from the internet by going to <http://www.oracle.com/technetwork/java/javase/downloads/index.html> and clicking on the 'download'

Academic Overview

button in the JDK column.

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 Resources

Useful Software

Whilst the JDK provides the compiler and runtime interpreter for the Java language, you will most likely want to make use of an Integrated Development Environment (IDE). You may use any IDE that you are comfortable with, but we strongly suggest you use **Eclipse**, because of the features included within it. Eclipse can be downloaded from: <http://www.eclipse.org/>

Another piece of software which you may like to install is *TortoiseSVN*, which is an implementation of *subversion* that works on windows. (Subversion was originally a Unix/Linux program). 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). This tool may be explained during the semester. Its main purpose is to keep track of the changes made to a set of files (a directory hierarchy), by storing snapshots of the file contents into a database, so that if you need to you can get back an older version of a file. (It is also helpful if you are working in a team situation where everyone needs to share the same copy of files).

Examination material or equipment

It is a closed book exam. No material or equipment besides pens/pencils is permitted.

Unit Schedule

| Week | Activities | Assessment |
|------|---|---|
| 0 | Register for tutorials and check out the unit website, review what you learned in FIT1002 | No formal assessment or activities are undertaken in week 0 |
| 1 | Topic 1: Review of Java Syntax and Basic Concepts | No practical in Week 1 |
| 2 | Topic 2: Object Orientation | Formative assessment item (topic 1) |
| 3 | Topic 3: Association and Aggregation Relationships | Commence first assessed practical |
| 4 | Topic 4: Inheritance and Polymorphism | Assessed Practical (topics 2 and 3) due |
| 5 | Topic 5: Interfaces, Abstract Classes and Callbacks | Assessed Practical (topic 4) |
| 6 | Topic 6: File Input and Output, and Exceptions | Formative assessment item (topic 5) |
| 7 | Topic 7: Data Structures and the Java Collection Classes | Assessed Practical (topic 6) |
| 8 | Topic 8: Recursion | Assessed Practical (topic 7) |
| 9 | Topic 9: Graphical User Interfaces and Event Handling | Assessed Practical (topic 8) |
| 10 | Topic 10: Algorithms for Searching and Sorting | Assessed Practical (topic 9) |
| 11 | Topic 11: Testing and Debugging | Formative assessment item(topic 10) |
| 12 | Revision | Formative assessment item (topic 11); Major Assignment Due 11:59PM on Sunday 23 October 2011 |
| | SWOT VAC | No formal assessment is undertaken SWOT VAC |
| | Examination period | LINK to Assessment Policy: http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html |

*Unit Schedule details will be maintained and communicated to you via your MUSO (Blackboard or Moodle) learning system.

Assessment Requirements

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:

Topic 2 and 3 Practical: Simple Object Oriented program involving associations and aggregations

Description:

The Practical exercises listed in weeks 3 and 4, which covers topics 2 and 3, will be assessed. The work must be the result of your own individual efforts, with guidance given by your tutor (through answering questions you may have), or asked on the discussion forum.

Weighting:

5%

Criteria for assessment:

Broadly, the criteria used to assess your work will be:

1. Your ability to use basic object-oriented constructs to create a small multiple class program that compiles and executes without errors.
2. Your ability to explain how various object-oriented constructs have been incorporated into your program.
3. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

Week 4

Remarks:

You must commence working on the tasks prior to turning up to the class, because you must submit before the conclusion of the class. You should aim to be finished before the final half-hour of the class time, so that the tutor can mark you.

• **Assessment task 2**

Title:

Topic 4: Inheritance and Polymorphism - Practical

Description:

The practical exercises relating to topic 4 (Inheritance and Polymorphism) will be assessed.

Weighting:

4%

Criteria for assessment:

Broadly, the criteria used to assess your work will be:

1. Your ability to use programming constructs to demonstrate an understanding of inheritance and polymorphism.
2. Your ability to explain how you used the programming constructs in your program.
3. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

Week 5

Remarks:

You must commence working on the tasks prior to turning up to the class, because you must submit before the conclusion of the class. You should aim to be finished before the final half-hour of the class time, so that the tutor can mark you.

• **Assessment task 3**

Title:

Topic 6: File Input and Output - Practical

Description:

The practical exercises relating to topic 6 (File input and output) will be assessed.

Weighting:

4%

Criteria for assessment:

Broadly, the criteria used to assess your work will be:

1. Your ability to use programming constructs to demonstrate an understanding of file-based input and output.
2. Your ability to explain how you used the programming constructs in your program.
3. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

Week 7

• **Assessment task 4**

Title:

Topic 7: Data Structures and the Java Collection Classes - Practical

Description:

The practical exercises relating to topic 7 will be assessed.

Weighting:

4%

Criteria for assessment:

Assessment Requirements

Broadly, the criteria used to assess your work will be:

1. Your ability to use programming constructs to demonstrate an understanding of data structures and the Java Collection classes
2. Your ability to explain how you used the programming constructs in your program.
3. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

Week 8

• Assessment task 5

Title:

Topic 8: Recursion - Practical

Description:

The practical exercises relating to topic 8 (Recursion) will be assessed.

Weighting:

4%

Criteria for assessment:

Broadly, the criteria used to assess your work will be:

1. Your ability to use recursion in programming.
2. Your ability to explain how the elements of recursive solutions have been incorporated into your work.
3. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

Week 9

• Assessment task 6

Title:

Topic 9: Graphical User Interfaces - practical

Description:

The practical exercises relating to topic 9 (Graphical User Interfaces) will be assessed.

Weighting:

4%

Criteria for assessment:

Broadly, the criteria used to assess your work will be:

1. Your ability to use the various classes of the Java API relating to the creation of a graphical user interface involving frames, buttons, textfields, and similar components.
2. Your ability to write event-handling code to respond to user-initiated events in a graphical user interface.
3. The ability to manipulate the contents of graphical components.
4. Your ability to explain how you used the programming constructs relating to GUIs and event handling in your program.
5. Whether the program meets the behavioural requirements as specified.

Details on how grades are allocated will be explained in the specification.

Due date:

• **Assessment task 7**

Title:

Major Assignment

Description:

In addition to the assessment of the preceding practical classes, there is a major assignment which will integrate concepts from many of the topics of this unit. The program will involve sorting and searching, graphical user interactions, inheritance and polymorphism, and file input and output.

Weighting:

15%

Criteria for assessment:

More-specific criteria will be provided with the task specification document, but broadly, the criteria used to assess your work will include:

1. Your ability to use inheritance and polymorphism
2. Your ability to use association/aggregation
3. Your ability to use techniques of file input and output
4. Your ability to implement searching and sorting algorithms
5. Your ability to construct a functioning graphical user interface
6. Your ability to follow industry standards in terms of documenting your programs.
7. Your ability to explain how used the programming constructs in your program.
8. Whether the program meets the behavioural requirements as specified.
9. Your ability to perform problem solving to create a working solution from a given problem description.

Details on how grades are allocated will be explained in the specification. A program which does not compile without errors will not be given a pass or higher grade. A program which achieves all functional requirements but without using the expected programming constructs will not get a grade higher than Credit.

Due date:

Sunday 23 October 2011

Remarks:

You will not be given time during class to do this assignment. You will be able to start working on this assignment before the mid-semester break, although some aspects will not be taught until after the mid-semester break.

Examinations

• **Examination 1**

Weighting:

60%

Length:

3 hours

Type (open/closed book):

closed book

Electronic devices allowed in the exam:

None

Assignment submission

It is a University requirement

(<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html>) for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at <http://www.infotech.monash.edu.au/resources/student/forms/>. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an online quiz).

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

Other Information

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at:

<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>)
- Codes of Practice for Teaching and Learning
(<http://www.policy.monash.edu.au/policy-bank/academic/education/conduct/suppdocs/code-of-practice-tea>)

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 Guide** modules
- Weekly **lecture slides** and any **sample programs** used during lectures

Other Information

- Weekly practical tasks and solutions
- Assignment specification (and later a solution)
- Links to additional electronic resources (such as Java API documentation)
- Discussion forums
- The FIT2034 web site on Moodle, where most of the above resources can be located
- Lectures from some campuses are recorded and available for access at <http://mulo.monash.edu>

READING LIST

Recommended Other Text

Arnold, K., et al (2006), "The Java Programming Language", 4th edition, Sun Microsystems/Addison-Wesley. (ISBN: 0-321-34980-6) - *this is only useful as a reference if you want to find out more than what we teach.*