



MONASH University
Information Technology

FIT5151
Object-oriented business application development

Unit Guide

Semester 2, 2012

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: 21 Jun 2012

Table of Contents

<u>FIT5151 Object-oriented business application development - Semester 2, 2012</u>	1
<u>Mode of Delivery</u>	1
<u>Contact Hours</u>	1
<u>Workload</u>	1
<u>Unit Relationships</u>	1
<u>Prerequisites</u>	1
<u>Chief Examiner</u>	1
<u>Campus Lecturer</u>	1
<u>Caulfield</u>	2
<u>Academic Overview</u>	3
<u>Outcomes</u>	3
<u>Graduate Attributes</u>	3
<u>Assessment Summary</u>	4
<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>	5
<u>Prescribed text(s)</u>	5
<u>Recommended Resources</u>	5
<u>Recommended text(s)</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>	8
<u>Examination 1</u>	8
<u>Assignment submission</u>	8
<u>Online submission</u>	8
<u>Extensions and penalties</u>	8
<u>Returning assignments</u>	8
<u>Other Information</u>	9
<u>Policies</u>	9
<u>Student services</u>	9
<u>Reading list</u>	10

FIT5151 Object-oriented business application development - Semester 2, 2012

FIT5151 will aim at capitalising on what students have learned in [FIT9017](#) Foundations of programming (or equivalent). The unit covers more in-depth material to enable students to build business applications that follow good Software Engineering principles of maintainability, reusability and expandability. The emphasis will be on helping students acquire solid object-oriented programming knowledge and skills for building business applications. Popular object-oriented design patterns will be introduced whenever appropriate to illustrate effective design process in building larger systems.

Mode of Delivery

Caulfield (Evening)

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

Tutorial/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

Prerequisites

[FIT9017](#) or equivalent

Chief Examiner

[Dr Chris Ling](#)

Campus Lecturer

Caulfield

Chris Ling

Academic Overview

Outcomes

At the completion of this unit, students will have -

A knowledge and understanding of:

- how to produce well-run, well-tested and well-documented object-oriented software by following solid software engineering principles of maintainability, reusability and expandability;
- effective use of popular object-oriented design patterns in the design process of larger systems;
- how to effectively and efficiently develop object-oriented application solutions to business-related problem specifications.

Developed attitudes that enable them to:

- appreciate the responsibility of coming up with well-tested and documented programs;
- appreciate the need to maintain ethical conducts when programming by making sure the code used my program is their own or taken from a legitimate source with full acknowledgement.

Gained practical skills to:

- navigate around in an Integrated Development Environment in order to efficiently produce quality applications;
- develop good software testing strategies.

Demonstrated the communication skills necessary to:

- work in a team to come up with an integrated business software solution

explain their design and testing strategies in writing and in person through interviews.

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
Assignment 1	20%	Week 7, 11.59pm Sunday 9 September 2012
Assignment 2	20%	Week 12, 11.59pm Sunday 21 October 2012
Examination 1	60%	To be advised

Teaching Approach

Lecture and tutorials or problem classes

Lectures are used to present new 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
- Other: Staff responses to queries posted on 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

Based on previous student evaluations, we decided:

1. To provide more examples in the lecture to help with the understanding of concepts.
2. To improve the sequencing of certain topics.
3. To focus on programming skills rather than the implementation of prototype features in all practical assignments.

4. To align the assessment requirements with students' understanding of concepts rather than prototype features. The alignment will be done based on weekly learning objectives.

If you wish to view how previous students rated this unit, please go to <https://emuapps.monash.edu.au/unitevaluations/index.jsp>

Required Resources

Please check with your lecturer before purchasing any Required Resources. Limited copies of prescribed texts are available for you to borrow in the library, and prescribed software is available in student labs.

Prescribed Software

You must have the Java SE 6 (or later) 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' button in the JDK column.

Prescribed text(s)

Limited copies of prescribed texts are available for you to borrow in the library.

Barnes and Kolling. (2011). *Objects First with Java: A Practical Introduction Using BlueJ*. (5th) Prentice Hall (ISBN: 9780132835541).

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. Examples of IDE are BlueJ, Eclipse, JCreator and NetBeans.

Recommended text(s)

Stuart Reges and Marty Stepp. (2011). *Building Java Programs: A Back to Basics Approach*. (2nd) Addison Wesley (ISBN: 0-136-09181-4).

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Programming Concepts and Java - variables, operators, expressions, control structures, recursion, class structure, collections, primitive types, object types, exceptions, I/O, file I/O	Tutorial labs commence in Week 1 of semester. This week's exercise is a revision of prerequisite knowledge.
2	Object oriented Concepts - classes, objects, methods, attributes, message passing, constructors, inheritance, polymorphism, encapsulation, visibility, abstraction, packages, interacting classes, association, aggregation, composition	
3	Inheritance - subclasses, subtyping, substitution, overriding, types of inheritance, access modifiers	
4	Inheritance - abstract classes, multiple inheritance, interfaces, inner classes, enumerations	
5	Testing, testing tools (JUnit)	
6	GUI - event handling, components, layout, AWT and Swing libraries	
7	Persistence: Java database connectivity, more file I/O	Assignment 1 due 11.59pm Sunday 9 September 2012
8	Program Design - design techniques (responsibility driven design), Parnas' principles, design representation (UML)	
9	Program design - coupling and cohesion, Law of Demeter, Design by Contract, Assertions, Refactoring	
10	Design Patterns - decorator, singleton, factory, observer, etc., frameworks	
11	Software development methodologies, agile methods	
12	Bringing it all together--Revision	Assignment 2 due 11.59pm Sunday 21 October 2012
	SWOT VAC	No formal assessment is undertaken in 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

Faculty Policy - Unit Assessment Hurdles

(<http://www.infotech.monash.edu.au/resources/staff/edgov/policies/assessment-examinations/unit-assessment-hu>)

Academic Integrity - Please see the Demystifying Citing and Referencing tutorial at

<http://lib.monash.edu/tutorials/citing/>

Assessment Tasks

Participation

- **Assessment task 1**

Title:

Assignment 1

Description:

The design and implementation of an application employing the principles introduced in the early to mid part of the semester.

Details will be made available in the assignment specification.

Weighting:

20%

Criteria for assessment:

This is an individual assignment and must be entirely your own work.

Submissions will be judged on their compliance with the stated functional requirements, code and design principles presented in the unit.

Assessment of this assignment *may be subject to an interview should the marker feels necessary.*

Due date:

Week 7, 11.59pm Sunday 9 September 2012

- **Assessment task 2**

Title:

Assignment 2

Description:

The design and implementation of an application employing the principles introduced in the mid to later part of the semester.

Details will be made available in the assignment specification.

Weighting:

20%

Criteria for assessment:

This is an individual assignment and must be entirely your own work.

Submissions will be judged on their compliance with the stated functional requirements, code and design principles presented in the unit.

Assessment Requirements

Assessment of this assignment *may be subject to an interview should the marker feels necessary.*

Due date:

Week 12, 11.59pm Sunday 21 October 2012

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).

Online submission

If Electronic Submission has been approved for your unit, please submit your work via the VLE site for this unit, which you can access via links in the my.monash portal.

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. For Sunway see <http://www.monash.edu.my/Student-services>, and for South Africa see <http://www.monash.ac.za/current/>

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. At Sunway, visit the Library and Learning Commons at <http://www.lib.monash.edu.my/>. At South Africa visit <http://www.lib.monash.ac.za/>.

Academic support services may be available for students who have a disability or medical condition. Registration with the Disability Liaison Unit is required. Further information is available as follows:

- Website: <http://monash.edu/equity-diversity/disability/index.html>;
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus, or Student Community Services Department, Level 2, Building 2, Monash University, Sunway Campus
- Telephone: 03 9905 5704, or contact the Student Advisor, Student Community Services at 03 55146018 at Sunway

Reading list

The following may provide useful extra reading for this unit. Copies of these are available in the Library (on reserve, one day loan or in the normal circulation).

Java Foundations, Lewis, De Pasquale & Chase, (Pearson Education), 2008

Big Java (4th edition), Cay Horstman (John Wiley & Sons), 2010

Java Programming - from Problem Analysis to Program Design (3rd edition), D. S Malik (Thomson), 2008

Thinking in Java (4th edition), Eckell (Prentice Hall), 2006

Absolute Java (3rd edition), Savitch (Addison Wesley), 2008