FIT1002
Computer programming

Unit Guide

Semester 1, 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: 22 Feb 2012

Table of Contents

FIT1002 Computer programming - Semester 1, 2012

This unit will provide students with an overview of programming and its role in problem-solving and strategies for meeting user requirements and for designing solutions to programming problems. The fundamental programming concepts of the memory model, data types, declarations, expressions and statements, control structures, block structure, modules, parameters and input and output will be applied within the context of objects, attributes, methods, re-use, information-hiding, encapsulation and message-passing. Software engineering topics include maintainability, readability, testing, documentation and modularisation.

Mode of Delivery

- Caulfield (Day)
- Clayton (Day)
- Gippsland (Day)
- Gippsland (Off-campus)
- Sunway (Day)
- South Africa (Day)

Contact Hours

2 hrs lectures/wk, 2 hrs laboratories/wk, 1 hr tutorial/wk

Workload

For on campus students, workload commitments are:

- two hours of lectures
- one-hour tutorial class in a flat room without computers to discuss theoretical programming concepts and develop problem solving strategies
- two-hour laboratory (practical class) (requiring advance preparation)
- a minimum of 2-3 hours of personal study **per one hour of lecture time** in order to satisfy the reading, tute, prac and assignment expectations.

Off-campus students generally do not attend lecture, tutorial and laboratory sessions, however, you should plan to spend equivalent time working through the relevant resources and participating in discussion groups each week.

Unit Relationships

Prohibitions

CPE1001, CSE1202, GCO1811, MMS1801, MMS1802

Chief Examiner

Mr Stephen Huxford

Campus Lecturer

Caulfield

Stephen Huxford

Clayton

Prof David Green

Gippsland

Shane Moore

South Africa

Sheelagh Walton

Sunway

Mylini Munusamy

Academic Overview

Outcomes

At the completion of this unit students will have -
An understanding of:

- the relationship between a problem description and program design;
- the management of problems using recognised frameworks;
- the use of design representations;
- the semantics of imperative programs;
- the object oriented paradigm as represented by Java;
- the sequence of steps that a computer takes to translate source code into executable code; and
- primitive data types and basic data structures.

Developed attitudes that enable them to:

- adopt a problem-solving approach;
- recognise the importance of programming and documentation conventions;
- appreciate quality parameters in program development;
- accept the code of professional conduct and practice; and
- act in accordance with best practice, industry standards and professional ethics.

Developed the skills to:

- use diagrams to design solutions for programming problems;
- apply problem solving strategies;
- use pseudo-code to design algorithms;
- design object oriented solutions to simple problems using multiple user-defined classes;
- create and test programming solutions to problems using the Java programming language;
- edit, compile and execute a computer program;
- analyse and debug existing programs; and
- write a test plan.

Demonstrated the communication skills necessary to:

- produce formal documentation for a program; and
- explain an existing program.

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 - JAVA basics	10%	Friday 5:00pm EST Week 7
Assignment 2 - Designing a JAVA application involving several classes and array of objects	10%	Friday 5:00pm EST Week 11
Laboratory Work and Short Tutorial Tests	20%	The end of the Laboratory session in weeks 2 - 11 in which the work is performed.
Examination 1	60%	To be advised

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
   • Test results and feedback
   • Quiz results
   • Solutions to tutes, labs and assignments

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

Improvements we have made to the unit from last delivery include:

1. Additions to lecture notes to include alternate explanantions of some concepts

2. Redevelop some of the tutorial and laboratory exercises to focus on activities directly related to the lecture topics of each week

3. Down-size final assignment in difficulty and marks allocated

4. Inclusion of a very short weekly test (with allocated marks) to ensure students are aware of the main points covered in lectures

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. Prescribed texts are available for you to borrow in the library, and prescribed software is available in student labs.

Prescribed Text:
Java Foundations, Lewis, DePasquale and Chase 2nd edition. Pearson 2011,
ISBN 10: 013705534X / 0-13-705534-X
ISBN 13: 9780137055340

Java Development Kit, Version 6 or later, published by Oracle. The latest version (Version 7) is available for download for free from: http://docs.oracle.com/javase/7/docs/webnotes/install/index.html

In addition to the Java Development Kit, students should have access to at least one of the following Integrated Development Environments, which the FIT1002 teaching team recommend:

- **Eclipse** (latest version) downloaded from http://eclipse.org.

- **JCreator** (latest version) downloaded from http://www.jcreator.com/
  Students are advised to download the free version (i.e. the LE version). There is no need for the fuller facilities provided in JcreatorPro

- **jEdit** (latest version) downloaded from http://www.jedit.org/

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Unit Administration and Introduction	No Tutes or Labs Students should familiarise themselves with their lab and the lab software prescribed for their campus (e.g. Eclipse, JCreator etc)
2	Algorithms, Variables and Data Types	Marked Tutes and Labs begin
3	Using objects and classes, Math Class, String Class, Random Class and I/O	
4	Selection	Assignment 1 Specification released
5	Repetition	
6	Modularisation	
7	Classes and Objects	Assignment 1 due Friday 5:00pm EST
8	Methods revisited	Assignment 2 Specification released
9	Object references	
10	Arrays	
11	Case study: Multiple classes	Marked Tutes and Labs end. Assignment 2 due Friday 5:00pm EST
12	Campus specific week	Assignment 2 Interviews Held
	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

Faculty Policy - Unit Assessment Hurdles
(http://www.infotech.monash.edu.au/resources/staff/edgov/policies/assessment-examinations/unit-assessment-hu

Assessment Tasks

Participation

- **Assessment task 1**

    Title:
    Assignment 1 - JAVA basics
    Description:
    This assignment will aim to help you to develop programs in Java using classes and
    objects that are able to read input from the user and format output. You will also learn to
    use classes and their methods that are available from the Java library.
    Weighting:
    10%
    Criteria for assessment:
    Detailed assessment criteria will be issued along with the assignment.

    1. All programs must run and compile correctly. Evidence of testing is required.
    2. Programs must meet the problem specification
    3. JAVA code should be readable and maintainable and follow the style
       recommended in the prescribed text book.
    4. Programs should be documented
    5. Students should be able to answer questions about their own work

    Due date:
    Friday 5:00pm EST Week 7

- **Assessment task 2**

    Title:
    Assignment 2 - Designing a JAVA application involving several classes and array of
    objects
    Description:
    This assignment will require students to use the selection and iteration control structures
    and methods. Students will also be expected to design UML class diagrams and write
    Java code to solve a problem that will involve many classes and an array of objects.
    Weighting:
    10%
    Criteria for assessment:
    Detailed assessment criteria will be issued along with the assignment.

    1. All programs must run and compile correctly. Evidence of testing is required.
    2. Programs must meet the problem specification
    3. JAVA code should be readable and maintainable and follow the style
       recommended in the prescribed text book.
    4. Programs should be documented

> 5. Students should be able to answer questions about their own work during an interview scheduled outside the lab class

Due date:
> Friday 5:00pm EST Week 11

- **Assessment task 3**

  Title:
  > Laboratory Work and Short Tutorial Tests

  Description:
  > In week 2 - 11 students will be expected to write and execute code to perform a task specified at the start of their Laboratory session. The specified coding task will come from a Laboratory task specification sheet released prior to each Laboratory session allowing for preparation.
  >
  > Students will be expected to complete a very short, closed book test at the start of each Tutorial session. The test will ask 5 short answer questions on the lecture material for the week the Tutorial is based on.

  Weighting:
  > 20%

  Criteria for assessment:
  > Laboratory work will be assesed during the Laboratory session. Full marks will require both working code and good coding style with the latter carrying more weight.
  >
  > Tutorial short tests will be marked by tutors after the tutorial. The questions will examine both conceptual and syntax knowledge covered in the lecture slides relevant to the tutorial. Required answers will be short and generally attract a mark of 0 or 1 (scaled appropriately for inclusion into the final mark for the unit).

  Due date:
  > The end of the Laboratory session in weeks 2 - 11 in which the work is performed.

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-po
- 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 Commuity Services at 03 55146018 at Sunway

Other

READING LIST

For all students:

Malik D.S., *Java Programming - From Analysis to Design.*, Thomson Learning 2006, ISBN: 0619216085

Robertson LA, *Simple Program Design*, 5th ed., Thomson/Nelson, 2007, ISBN: 017010704-3

 For student with advanced programming skills:

Arnold K., Gosling J. & Holmes D., *The Java Programming Language*, Fourth Edition, Addison-Wesley, Upper Saddle River, NJ, 2006.   ISBN: 0-321-34980-6 (paperback)