FIT9017
Foundations of programming

Unit Guide

Semester 1, 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: 23 Feb 2010
## Table of Contents

**FIT9017 Foundations of programming - Semester 1, 2010**

- Chief Examiner: ................................................................. 1
- Lecturer(s) / Leader(s): .......................................................... 1
  - Caulfield ................................................................. 1
  - Gippsland ............................................................. 1

### Introduction ........................................................................... 2
### Unit synopsis ......................................................................... 2
### Learning outcomes ............................................................... 2
### Contact hours ........................................................................ 2
### Workload .............................................................................. 2
### Unit relationships ................................................................... 3
  - Prohibitions ..................................................................... 3
### Teaching and learning method ............................................... 4
  - Teaching approach ......................................................... 4
  - Timetable information .................................................... 4
  - Tutorial allocation .......................................................... 4
  - Off-Campus Learning or flexible delivery ............................ 4
  - Unit Schedule ............................................................... 4

### Unit Resources ......................................................................... 6
  - Prescribed text(s) and readings ....................................... 6
  - Recommended text(s) and readings ................................ 6
  - Required software and/or hardware .................................. 6
  - Equipment and consumables required or provided ................ 6
  - Study resources ............................................................. 6

### Assessment ............................................................................ 8
  - Overview .......................................................................... 8
  - Faculty assessment policy ............................................... 8
  - Assignment tasks ............................................................ 8
  - Examination ...................................................................... 9
  - Due dates and extensions ................................................ 9
  - Late assignment .............................................................. 9
  - Return dates .................................................................... 9

### Appendix .............................................................................. 10
FIT9017 Foundations of programming - Semester 1, 2010

Chief Examiner:

Dr Judithe Sheard
Senior Lecturer
Phone: +61 3 990 32701
Fax: +61 3 990 31077

Contact hours: To be advised

Lecturer(s) / Leader(s):

Caulfield

Dr Judithe Sheard
Senior Lecturer
Phone: +61 3 990 32701
Fax: +61 3 990 31077

Contact hours: To be advised

Gippsland

Dr Shyh Wei Teng
Lecturer
Phone: +61 3 990 26851

Contact hours: To be advised
Introduction

Welcome to FIT9017 Foundations of Programming for semester 1, 2010. This 6 point unit is a core unit for the Master of Applied Information Technology. Computer programming is a fundamental activity underlying all computer systems. A knowledge of programming is therefore important for students interested in a future in information technology.

Unit synopsis

This unit aims to provide students with the basic concepts involved in the development of well structured software using a programming language. It concentrates on the development of problem solving skills applicable to all stages of the development process. Students gain experience with the translation of a problem specification into a program design, and the implementation of that design into a programming language. The subject introduces software engineering topics such as maintainability, readability, testing, documentation, modularisation, and reasoning about correctness of programs. Students are expected to read and understand existing code as well as develop new code.

Learning outcomes

At the completion of this unit students will:

- be competent in designing, constructing, testing and documenting small computer programs using Java;
- be able to demonstrate the software engineering principles of maintainability, readability, and modularisation; and,
- understand the concepts of the object-oriented style of programming.

Contact hours

2 hrs lectures/wk, 2 hrs laboratories/wk

Workload

The workload commitments are:

On-Campus Students

two-hour lecture each week; two-hour tutorial each week held in a computer laboratory; anda minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

Off-Campus-Learning (OCL) Students

although no face-to-face classes are scheduled, OCL students are required to spend ~12 hours per week working through the weekly study materials on the unit MUSO site and participating in the discussion groups each week.
Unit relationships

Prohibitions

CSE9000
Teaching and learning method

Teaching approach

For On-Campus students, this unit will be delivered via a 2 hour lecture and a 2 hour tutorial class each week.

Lectures will be used to present and explain programming constructs and the fundamentals of program design and testing.

Tutorials will be used for practical experience in the design, coding, testing and debugging of programs.

For Off-Campus-Learning students, the curriculum is defined by what is covered in the Study Guide modules for each week.

Tutorials will also be used for practical experience in the design, coding, testing and debugging of programs.

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-Learning students should refer to the study guide modules as their primary source of direction for studying in this unit.

Unit Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Date*</th>
<th>Topic</th>
<th>References/Readings</th>
<th>Key dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01/03/10</td>
<td>Intoduction to FIT9017 and expectations, introduction to programming, basic OO concepts, objects, classes, attributes, behaviour, state and identity.</td>
<td>Chapter 1, Barnes &amp; Kolling</td>
<td>Note: Lecture outline and due dates for assessment tasks may be subject to change as semester progresses</td>
</tr>
<tr>
<td>2</td>
<td>08/03/10</td>
<td>Class definition, fields, constructors, methods, parameter passing, variables, expressions, statements, assignment, assignment, expressions, primitive data types, Strings, basic output.</td>
<td>Chapter 2, Section 2.1-2.12, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15/03/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Topic</td>
<td>Chapter(s)</td>
<td>Assignment/Exercise</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>22/03/10</td>
<td>Selection (if and switch statements), conditions, arithmetic, relational &amp; logical operators, shorthand operators, precedence, scope and lifetime, basic input.</td>
<td>Chapter 2, Section 2.13-2.19, Barnes &amp; Kolling</td>
<td>Exercise on Plagiarism, Cheating and Collusion (Hurdle)</td>
</tr>
<tr>
<td>5</td>
<td>29/03/10</td>
<td>Object creation and interaction, abstraction, modularisation, class &amp; object diagrams, object creation, primitive vs. object types, method calling, message passing, method signatures, method overloading.</td>
<td>Chapter 3, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing, unit testing, regression testing, specifying a test strategy, debugging.</td>
<td>Chapter 6, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Mid semester break</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>12/04/10</td>
<td>Class libraries, importing classes, collections, ArrayLists, Arrays, iteration, pre and post test loops.</td>
<td>Chapter 4, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>19/04/10</td>
<td>Class documentation, javadoc, identity vs. equality, more on strings, sets and maps, conditional operator.</td>
<td>Chapter 5, Section 5.1-5.9, Barnes &amp; Kolling</td>
<td>Assignment 1 due (20%)</td>
</tr>
<tr>
<td>8</td>
<td>26/04/10</td>
<td>Information hiding, encapsulation, access modifiers, scoping, class variables, class methods, constants.</td>
<td>Chapter 5, Section 5.10-5.14, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>03/05/10</td>
<td>Program design, design documentation</td>
<td>Chapter 13, Section 13.1-13.3, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10/05/10</td>
<td>Code quality, coupling, cohesion, refactoring, testing a program.</td>
<td>Chapter 7, Barnes &amp; Kolling</td>
<td>Assignment 2 - Stage 1 due (4%)</td>
</tr>
<tr>
<td>11</td>
<td>17/05/10</td>
<td>Inheritance, superclasses, subclasses, subtypes, substitution, polymorphic variables, protected access, casting, wrapper classes, collection hierarchy</td>
<td>Chapter 8, Barnes &amp; Kolling</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>24/05/10</td>
<td>Programming errors, exception handling, file I/O</td>
<td>Chapter 9, Barnes &amp; Kolling</td>
<td>Assignment 2 - Stage 2 due (26%)</td>
</tr>
<tr>
<td>13</td>
<td>31/05/10</td>
<td>Revision, exam discussion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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.*
Unit Resources

Prescribed text(s) and readings


This is the textbook for the unit. The course will follow this text. The text contains the weekly pre-reading and many exercises that will be specified for you to work on in the tutorial classes and outside class.

The text book is available from the Monash University Bookshop at the Caulfield campus.

Recommended text(s) and readings

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


*Big Java* (3rd edition) by Cay Horstman (John Wiley & Sons), 2008

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


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

Required software and/or hardware

In this unit we will use Java and the BlueJ development environment. This software is available on CD with the text book.

Also:

The Java software is available to download from Sun website at: [http://java.sun.com/javase/downloads/](http://java.sun.com/javase/downloads/).

BlueJ is available to download from the BlueJ site at: [http://www.bluej.org/](http://www.bluej.org/). You will be given instructions on how to use this in your first tutorial. You are expected to work in the BlueJ development environment. Tutors will only assess the assignments under this environment.

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.

Study resources

Study resources we will provide for your study are:

- Weekly lecture notes;
Weekly tutorial exercises;
Weekly readings from the text book;
Assignment specifications;
A sample examination;
This Unit Guide outlining the administrative information for the unit;
The unit web site on MUSO, where resources outlined above will be made available. This site will be updated at least each week so you should access it regularly.

For Off-Campus-Learning students, weekly study guides modules will also be made available on the unit site.
Assessment

Overview

Examination (3 hours): 50%; In-semester assessment: 50%

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.

The unit is assessed with two assignments, one class test and a three hour closed book examination.

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 task 1

  Title:  
  Exercise on Plagiarism, Cheating and Collusion

  Description:  
  Students will complete exercises in class to make sure they are familiar with and fully understand the concepts, rules and issues relating to plagiarism, cheating and collusion with respect to work submitted for assessment in this unit

  Weighting:  
  Hurdle

  Due date:  
  Conducted in Week 4 tutorial classes.

• Assignment task 2

  Title:  
  Assignment 1, Assignment 2 - Stage 1 & Assignment 2 - Stage 2

  Description:  
  These assignments will require students to design, write, test and document a program in Java.
Weighting:
20%, 4% & 26% respectively

Due date:
Assignment 1 - Week 7; Assignment 2 (Stages 1 and 2) - Weeks 10 & 12 respectively.
Dates to be advised

Examination

- Weighting: 50%
- Length: 3 hours
- Type (open/closed book): Closed book

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 NOT to assume that granting 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 submission deadline will be subject to a penalty of 10% per day of the possible marks.

Return dates

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