

Unit Title – 7165

Unit Outline 2009

Faculty of Information Sciences and Engineering

University of Canberra

*Australian Government Higher Education (CRICOS)
Registered Provider number: #00212K*

Unit Title – 7165

Unit Outline 2009

Faculty of Information Sciences and Engineering

University of Canberra

*Australian Government Higher Education (CRICOS)
Registered Provider number: #00212K*

This Unit Outline must be read in conjunction with:

- a) *Studying at the University of Canberra: A Guide to Policies and Procedures*, which sets out University-wide policies and procedures, including information on matters such as plagiarism, grade descriptors, moderation, feedback and deferred exams, and is available at <http://www.canberra.edu.au/student-services>
- b) *Guide to Student Services at the University of Canberra*, and is available at <http://www.canberra.edu.au/student-services>
- c) Any additional information specified in section 6f.

1: General Information

1a Unit title: Object-Oriented Software Design

1b Unit number: 7165

1c Semester and year offered: Semester 1, 2009

1d Credit point value: 3 CP

1e Unit level: UG – Third Year

1f Name of Unit Convener and contact details (including telephone and email)

Unit Convenor: Bala M. Balachandran
Room: 11C22
Phone: (02) 6201 2622
Email: bala.balachandran@canberra.edu.au

1g Administrative contact details (including name, location, telephone and email)

Office contact: School Office
Room: 11B14
Phone: (02) 6201 2417
Fax : (02) 6201 5231
Email: ise@canberra.edu.au

2: Academic Content

2a Unit description and learning outcomes

Unit Description: This unit is based on object-oriented techniques applied to software engineering. It covers a range of areas including current software development techniques and methodologies, software life cycles, requirements engineering, O-O modelling with UML 2.0, software architecture, system design, object design, design patterns, mapping models to code, mapping objects to databases, full cycle testing, TDD, and refactoring. The unit is structured around weekly lectures that introduce the theory component and a tutorial program that is based on small group work where students practice their Object-Oriented skills and knowledge using a case study. All practical work will be carried out using the .NET platform and C#.

Learning Outcomes: On successful completion of this you will be able to:

1. Employ agile software development (ASD) using object-oriented and relational database technologies;
2. Apply Agile Model Driven Development (AMDD) approach to requirements, analysis and design techniques applying the diagrams of the Unified Modelling Language (UML 2) as well as leading-edge modelling techniques;
3. Perform a range of modelling such as usage modelling, conceptual domain modelling, process modelling, structural design modelling and dynamic object design to your software project;
4. Describe a collection of techniques that can be used throughout the entire software development lifecycle (SDLC), including unit testing techniques that you can apply as part of your TD efforts;
5. Explain software architectural styles and patterns as well as several modelling techniques that you can apply to explore the high-level design of your system;
6. Master Agile Object Programming techniques including TDD and refactoring;
7. Develop skills of applying design patterns in the development of your system;
8. Understand the existing and emerging technologies that enable you to develop distributed object-based systems

2b Prerequisites and/or co-requisites

Software Technology 2 or Software Engineering 2A

And

Information Systems Analysis and Modelling or Systems Analysis and Modelling

3: Delivery of Unit and Timetable

3a Delivery mode

This unit will be delivered in traditional mode that is on-campus in standard semesters with weekly lectures/tutorials/laboratory classes:

3b Schedule of topics/lectures/tutorials/practicals/field classes by week

Lectures

Week	Activity
1	Unit Overview Software Development Methodologies – A Review Agile Software Development
2	O-O Modelling with UML 2.0 – An Overview Usage Modelling Conceptual Domain Modelling Business Process Modelling
3	Dynamic Object Design and Class Diagrams GRASP - General Responsibility Assignment Software Patterns
4	Software Architecture Common Architectural Styles
5	Software Architectural Modelling with UML 2.0
6	Applying Design Patterns
7	Full Lifecycle Object-Oriented Testing (FLOOT) Test-Driven Development using NUnit
8	Mid-semester break
9	Mid-semester break
10	Design Patterns II
11	Persistence Design and Development User Interface Development
12	Introduction to Refactoring Refactoring Techniques
13	Refactoring – A Case Study
14	Distributed Object Technologies
15	Unit Summary and Exam Overview

Tutorial/Laboratory Classes:

Week	Activity
1	No Class
2	Forming Teams, Introduction to Project and Assignment 1
3	Requirement Analysis, Usage Modelling Business Process Modelling – Groups 1, 2
4	Conceptual Domain Modelling – Groups 3, 4
5	Interaction Diagrams Class Diagram – Groups 5, 6
6	Final Design Presentation 1 - Groups 1, 3 and 5
7	Final Design Presentation 2 - Groups 2, 4 and 6 Assignment 1 Submission
8	Mid-semester break

9	Mid-semester break
10	TDD using NUnit
11	Databases and C#
12	NUnit Demo – Members of Groups 2, 4 and 6
13	NUnit Demo – Members of Groups 1, 3 and 5
14	Final Project Presentation – Groups 2, 4 and 6
15	Final Project Presentation – Groups 1, 3 and 5

4: Unit Resources

4a Lists of required texts/readings

Major Reference Book:

- Larman, C. *Applying UML and Patterns, Introduction to OOAD and Iterative Development* (3rd Ed, 2005), P/H

Other complimentary reference books:

- Ambler, S.W. *The Object Primer*, 3rd Edition, Cambridge University Press, 2004
- Stumpf, R.V and Teague, L.C. *Object-Oriented Systems Analysis and Design with UML*, 2005, P/H
- Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (2005), A/W
- Fowler, M. *Refactoring: Improving the design of existing code* (2001), A/W
- Deitel, et al, *C# How to Program* (2005), P/H

4b Materials and equipment

All assignments are programming assignments. These require access to a Win32 based computer (Windows NT or 2000) and the .NET environment. These environments are provided in Building 11 laboratories.

4c Unit website

The URL for the unit web site is:

<http://www.isc.canberra.edu.au/un7165/>

The URL for the University of Canberra library home page is:

<http://www.canberra.edu.au/library/>

The URL for the University of Canberra online library catalogue is:

<http://library.canberra.edu.au/>

5: Assessment

5a Assessment overview

Assessment Item (including exams held in the exam period)	Due Date of Assignments	Weighting (Total to equal 100%)
O-O Analysis and Design Presentation	Tutorial Week 3/4/5	5%
Final System Design Presentation	Tutorial Week 6/7	5%
Assignment 1	Friday, Week 7	20%
Unit Testing Demonstration	Tutorial Week 12/13	5%
Final Project Demonstration	Tutorial Week 14/15	5%
Assignment 2	Friday, Week 15	20%
Final Examination	S1 Exam Period	40%

5b Details of each assessment item

Assignment 1: Tasks include O-O analysis and design using UML 2.0 and system design modelling and documentation.

Assignment 2: Tasks include TDD using NUnit, Refactoring, Mapping models to code, implementation of persistent using DBMS, applying design patterns, and coding using C#/NET.

Tutorial Presentations: Each student will give two oral presentations and two practical demonstrations during the semester.

Presentation 1– Requirements Elicitation and Usage Modelling/Conceptual Domain Modelling/Object Dynamic Modelling.

Presentation 2 – Software Architecture, System Design and Modelling

Demonstration 1– Unit Testing using NUnit

Demonstration 2 – Final Project Demo

Final Examination:

A three-hour examination will be held during the examination period; two A4 sides of handwritten or typed notes permitted. The examination is intended to explore your understanding of the key concepts, methodologies and techniques covered during the semester, and how you applied these in your project.

5c Special assessment requirements

Each assessment item will be given a grade and an associated percentage mark. The marks for the assignments, tutorial presentations/demos and final examination are totalled to produce an overall unit mark.

Overall unit Mark = Assignment 1(20%) + Assignment 2(20%) + Presentation 1(5%) + Presentation 2(5%)+ Demonstration 1(5%) + Demonstration 2(5%)+ Final Examination (40%)

In order to gain a **PASS GRADE** in this unit you **MUST** achieve minimum a **PASS GRADE (50%)** in each of the following assessments:

- **Assignment 1**
- **Assignment 2**
- **Final Examination**

Grades will be awarded as follows:

Grade	Overall Unit Mark
HD	85-100
DI	75-84
CR	65-74
P	50-64

Please note that the PX grade is no longer available.

5d Supplementary assessment

None

5e Text-matching software

None

6: Student Responsibility

6a Workload

The amount of time you will need to spend on study in this Unit will depend on a number of factors including your prior knowledge, learning skill level and learning style. Nevertheless, in planning your time commitments you should note that for a 3cp Unit the total notional workload over the fifteen-week semester is assumed to be 150 hours or an average of 10 hours per week. These hours include time spent in classes. The total workload for Units of different credit point value should vary proportionally. For example, for a 6cp Unit the total notional workload over a fifteen-week semester is assumed to be 300 hours or an average of 20 hours per week.

6b Special needs

Students who need assistance in undertaking the unit because of disability or other circumstances should inform their Unit Convener or the Disabilities Office as soon as possible so the necessary arrangements can be made.

6c Attendance requirements

6d Required IT skills

6e Costs

Students must expect to pay for the textbook.

6f Additional information

None

7: Student Feedback

All students enrolled in this Unit will have an opportunity to provide anonymous feedback on the Unit at the end of the Semester via the Unit Satisfaction Survey which will be presented to you on OSIS. Your lecturer or tutor may also invite you to provide more detailed feedback on their teaching through an anonymous in-class questionnaire administered through the University's Teaching and Learning Centre (TLC).

8: Authority of this Unit Outline

Any change to the information contained in Section 2 (Academic content), Section 3 (Delivery of Unit and timetable) and Section 5 (Assessment) of this document, will only be made by the Unit Convener if the written agreement of staff and a majority of students has been obtained; and if written advice of the change is then forwarded to each student enrolled in the Unit at their registered term address. Any individual student who believes him/herself to be disadvantaged by a change is encouraged to discuss the matter with the Unit Convener.