MACQUARIE
University
SYDNEY · AUSTRALIA

ITEC802

Object Oriented Technology

S2 Evening 2014

Computing

Contents

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff
Unit Convenor
Pongsak Suvanpong
pongsak.suvanpong@mq.edu.au
Contact via pongsak.suvanpong@mq.edu.au
E6A 330
TBA

Lecturer
Luke Mathieson
luke.mathieson@mq.edu.au
Contact via luke.mathieson@mq.edu.au
E6A 313
TBA

Credit points
4

Prerequisites
COMP333 or COMP344 or COMP355

Corequisites

Co-badged status

Unit description
Object-oriented design approaches apply methods for producing applications software that place correctness and reusability as principal aims. Its tools include object classes, inheritance, message passing and polymorphism. This unit examines object-oriented design and programming with applications to simulation.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

Learning Outcomes

On successful completion of this unit, you will be able to:

A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.

Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.

Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.

Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase

Awareness of the importance of correctness of software, and competence in source level debugging and unit testing. Includes testing software for correctness using JUnit testing and debugger.

General Assessment Information

1. Late work will not be accepted. Special Consideration will only be granted provided that the application includes a medical certificate or professional authority showing clearly that the student was ill or otherwise incapacitated. Disruption to Studies Policy can be found here http://www.mq.edu.au/policy/docs/disruption_studies/policy.html
2. All assignments should be handed in via the online system at http://learn.mq.edu.au/ by the time specified in the assignment description.
3. All work submitted must be readable and complied with Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Tasks

Name	Weighting	Due
Practical Submissions	8%	fortnightly
Assignment 1	12%	TBA
Assignment 2	20%	TBA
Final Exam	60%	TBA

Practical Submissions

Due: **fortnightly**
Weighting: **8%**

these practical submissions assess the students' understanding of weekly topics as we go along.

there are 4 submissions, worth 2% for each submission. submissions are released every other week. the 4 submissions are designed with the following topics in mind -

1. object oriented principles, basics of class design

2. design principles, JUnit testing

3. design patterns

4. design patterns

On successful completion you will be able to:

- A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase

Assignment 1

Due: **TBA**
Weighting: **12%**

Assignment 1 introduces students to object oriented design and programming including -

1. designing classes

2. determining class relationships

3. designing a solution that is most "suitable" in the context of the problem domain

The assignment is structured as a scenario based problem in which the students are required to identify and design individual classes incorporating the object oriented principles and design principles discussed in the unit. There will be a problem solving component, specific to the problem domain, that the students will be required to solve as well.

On successful completion you will be able to:

- A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.

Assignment 2

Due: **TBA**
Weighting: **20%**

applying design patterns to solve a problem using object oriented methodology.

students are required to identify the design pattens that will contribute to the solution of the problem given.

since there will be multilpe patterns applied simultaneously, unwrapping the complex scenario will be required duing solving the problem.

assignment 2 will also involve unit testing. in saying this, students will be required to add unit tests using JUnit testing for the classes they design to confirm that each class does what it is required to do.

the deliverable for this assignment, like assignment 1, is a completed java project submitted as an archive file.

On successful completion you will be able to:

- A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase
- Awareness of the importance of correctness of software, and competence in source level debugging and unit testing. Includes testing software for correctness using JUnit testing and debugger.

Final Exam

Due: **TBA**
Weighting: **60%**

assesses learning outcome topics in an examination setting.

On successful completion you will be able to:

- A strong grasp of Object Oriented Principles and Design Principles. Includes

encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase

Delivery and Resources

Delivery of this unit is structured as a 2-hour lecture followed by a 2-hour practical class.

Computer labs are made available to students for the practical class and the required software is installed before the start of the semester.

The resources you will be required to install on your home computer / laptops are -

1. Java Standard Edition / Java SE (current version at the time of writing this document is 7u25)

2. Eclipse IDE for Java Developeres (or you can use Java NetBeans/ IntelliJ if you prefer those)

There are **no** prescribed books since there is no one book that satisfactorily covers all the topics from this unit. However, the following books are recommended for the purpose of this unit, in order of my personal preference.

Recommended:

1. Head First Design Patterns Elisabeth Freeman, Eric Freeman, Bert Bates & Kathy Sierra [2004]. O'Reilly Media, Inc. ISBN 0596007124.

2. Design Patterns: Elements of Reusable Object-Oriented Software Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison Wesley. ISBN: 9780201633610

3. Refactoring: Improving the design of existing code Martin Fowler. Addison Wesley. ISBN: 9780201485677

4. Applying UML and patterns Craig Larman. Prentice Hall PTR; 3 edition (October 30, 2004). ISBN-10: 0131489062

5. Thinking in Java Bruce Eckel [2002]**.** Prentice-Hall. AVailable for free from http://www.mindview.net/Books/TIJ/

6. UML 2.0 in a Nutshell Dan Pilone & Neil Pitman [2005]. O'Reilly Media, Inc.; 2nd ed. Edition. ISBN 0596007957.

Unit Schedule

Week	Topic	Practical	Further readings
1	**Introduction (Lecture slides provided)** Object Oriented Methodology History of OO methodology Components of OO methodology Driving forces in OO methodology	Classes and Objects Constructors	Thinking in Java Pages 13 - 92
2	**Object Oriented Principles (Lecture slides provided)** Best practices of OO methology OO concepts such as encapsulation, inheritance, polymorphism Static vs. Dynamic binding resulting in polymorphism	Classes and Objects Mutators and Accessors	Thinking in Java Pages 209 - 310
3	**Design Principles (Lecture slides provided, http://oodesign.com/)** SOLID principles	Java Containers	Thinking in Java Pages 791 - 900
4	**Case Study**	Exception Handling	Thinking in Java Pages 443 - 502
5	**Object Oriented Design (Lecture slides provided)** Interaction Diagrams Sequence Diagrams (SD) Dynamic collaboration Diagrams (DCD) Comparison of SD and DCD	File Input Output	Thinking in Java Pages 901 - 1010
6	**JUnit testing (Lecture slides provided)** Correctness of software Unit testing Writing good unit tests	JUnit testing	http://www.vogella.com/articles/JUnit/article.html

7	**Debugging & Source Code Version Control** Type of errors Tools for debugging Source level debugging Tracing Tools for source code version control	Testing Debugging	
8	**Design Patterns - 1** Singleton Adapter	Design Patterns	
9	**Design Patterns - 2 & GUI** Observer Model View Controller (MVC) Events Driven Programming Swing GUI.	GUI and MVC	
10	**Design Patterns - 3** Factory Method Abstract Factory Composite	Design Patterns	
11	**Design Patterns - 4** Iterator Decorator Strategy	Design Patterns	
12	**Designed Pattern - 5 & Multithreading** State and Concurrency Basic Thread & Problems Race Condition & Dead Lock Synchronization Critical Section Barrier	Multithreading and Concurrency with state pattern	
13	**Revision**		

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central. Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy http://mq.edu.au/policy/docs/assessment/policy.html

Grading Policy http://mq.edu.au/policy/docs/grading/policy.html

Grade Appeal Policy http://mq.edu.au/policy/docs/gradeappeal/policy.html

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Disruption to Studies Policy http://www.mq.edu.au/policy/docs/disruption_studies/policy.html *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

In addition, a number of other policies can be found in the Learning and Teaching Category of Policy Central.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/support/student_conduct/

Staff-Student Liaison Committee

The Department has established a Postgraduate Staff-Student Liaison Committee to provide all students studying a Computing unit the opportunity to discuss related issues or problems with both students and staff.

The committee meets three times during the semester. The meeting dates will be communicated to the students once available.

If you have exhausted all other avenues, then you should consult the Director, Postgraduate coursework program (Dr. Peter Busch) or the Head of Department (Dr Christophe Doche (christophe.doche@mq.edu.au)). You are entitled to have your concerns raised, discussed and resolved.

Assessment policy

If you cannot complete a piece of work please see the convenor **before** the due date. Check also the special consideration policy. A more detailed description of each task is given below.

Assessment tasks explained

As the table under assessment tasks indicates, there will be 3 assessment tasks.

- Four individual practical submissions
- Two individual assignments.
- One final examination.

Your final grade will depend on your performance in each part separately. In particular:

- You must perform satisfactorily in the examination in order to pass this unit.
- You must submit a reasonable attempt the practical tasks and assignments to pass this unit.

Further, you need to get a minimum of following marks in the final exam to get corresponding grades -

- Pass - a minimum of 40% (24/60) in the final exam
- Credit - a minimum of 60% (36/60) in the final exam
- Distinction - a minimum of 70% (42/60) in the final exam
- High Distinction - a minimum of 80% (48/60) in the final exam

All assignments should be handed in via the online system at http://learn.mq.edu.au/ by the time specified in the assignment description.

All work submitted should be readable and well presented.

Late work will not be accepted, unless it is because of illness or other circumstances, in which case, please contact the lecturer **before** the due date.

Final Examination

For this unit, a final examination is fully appropriate to test learning and knowledge of all learning outcomes. In particular, It allows for accurate reflection of the degree of understanding of learning outcomes LO1, LO2, L03, L06, and L09.

The final examination accounts for 60% of the final mark. The 4 sections of the unit:

1. Object Oriented Principles
2. Design Principles
3. Object-Oriented Analysis and Design
4. Design Patterns

are equally weighted.

Regarding the examination process, note that

- the University Examination period for Semester 2, 2014 is from 17th November to 5 December 2014
- you are expected to present yourself for examination at the time and place designated in the University Examination Timetable
- the timetable will be available in Draft form approximately eight weeks before the commencement of the examinations and in Final form approximately four weeks before

the commencement of examinations

- no early examinations for individuals or groups of students will be set. All students are expected to ensure that they are available until the end of the teaching semester, that is the final day of the official examination period
- the only exception to not sitting an examination at the designated time is because of documented illness or unavoidable disruption. In these circumstances you may wish to consider applying for Special Consideration.

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

Student Services and Support

Students with a disability are encouraged to contact the Disability Service who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

IT Help

For help with University computer systems and technology, visit http://informatics.mq.edu.au/help/.

When using the University's IT, you must adhere to the Acceptable Use Policy. The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

PG - Discipline Knowledge and Skills

Our postgraduates will be able to demonstrate a significantly enhanced depth and breadth of knowledge, scholarly understanding, and specific subject content knowledge in their chosen fields.

This graduate capability is supported by:

Learning outcomes

- A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase
- Awareness of the importance of correctness of software, and competence in source level debugging and unit testing. Includes testing software for correctness using JUnit testing and debugger.

Assessment tasks

- Practical Submissions
- Assignment 1
- Assignment 2
- Final Exam

PG - Critical, Analytical and Integrative Thinking

Our postgraduates will be capable of utilising and reflecting on prior knowledge and experience, of applying higher level critical thinking skills, and of integrating and synthesising learning and knowledge from a range of sources and environments. A characteristic of this form of thinking is the generation of new, professionally oriented knowledge through personal or group-based critique of practice and theory.

This graduate capability is supported by:

Learning outcomes

- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase
- Awareness of the importance of correctness of software, and competence in source level debugging and unit testing. Includes testing software for correctness using JUnit testing

and debugger.

Assessment tasks

- Practical Submissions
- Assignment 1
- Assignment 2
- Final Exam

PG - Research and Problem Solving Capability

Our postgraduates will be capable of systematic enquiry; able to use research skills to create new knowledge that can be applied to real world issues, or contribute to a field of study or practice to enhance society. They will be capable of creative questioning, problem finding and problem solving.

This graduate capability is supported by:

Learning outcomes

- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Competence in identifying the need for design patterns and incorporating them in the system design. Includes design patterns including factory method, observer, abstract factory, iterator, decorator, strategy and state.
- Capability to design and program systems using object-oriented principles, design principles, and design patterns. Includes applying object oriented principles, design principles and design patterns during the implementation phase

Assessment tasks

- Practical Submissions
- Assignment 1
- Assignment 2
- Final Exam

PG - Effective Communication

Our postgraduates will be able to communicate effectively and convey their views to different social, cultural, and professional audiences. They will be able to use a variety of technologically supported media to communicate with empathy using a range of written, spoken or visual formats.

This graduate capability is supported by:

Learning outcome

- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.

Assessment tasks

- Assignment 2
- Final Exam

PG - Engaged and Responsible, Active and Ethical Citizens

Our postgraduates will be ethically aware and capable of confident transformative action in relation to their professional responsibilities and the wider community. They will have a sense of connectedness with others and country and have a sense of mutual obligation. They will be able to appreciate the impact of their professional roles for social justice and inclusion related to national and global issues

This graduate capability is supported by:

Learning outcome

- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.

Assessment tasks

- Practical Submissions
- Assignment 1
- Assignment 2

PG - Capable of Professional and Personal Judgment and Initiative

Our postgraduates will demonstrate a high standard of discernment and common sense in their professional and personal judgment. They will have the ability to make informed choices and decisions that reflect both the nature of their professional work and their personal perspectives.

This graduate capability is supported by:

Learning outcomes

- A strong grasp of Object Oriented Principles and Design Principles. Includes encaspulation, abstraction, inheritance, polymorphism, SOLID principles.
- Confidence in performing Object Oriented analysis and design. Includes performing use case analysis, using use case diagrams, and sequence diagrams.
- Awareness of the importance of correctness of software, and competence in source level debugging and unit testing. Includes testing software for correctness using JUnit testing

and debugger.

Assessment tasks

- Practical Submissions
- Assignment 1
- Assignment 2

Standards

Standards

Four standards, namely HD, D, CR, P summarize as many different levels of achievement. Each standard is precisely defined to help students know what kind of performance is expected to deserve a certain mark. The standards corresponding to the learning outcomes of this unit are given below:

HD	A strong grasp on object oriented principles and design principles, including the capability to identify issues with complex systems and the ability to solve the same.	Demonstrate a comprehensive capability to identify requirements of a system, document and design the system using object oriented methodology	Adequacy in using design patterns individually and in conjunction with others, based on requirements. Capability to identify "patterns diarrhea" at the same time and eliminate useless pattern application from the design	Capability to apply LO1 and LO3 in a technical environment. Complete command over programming environments and platforms available. Can adapt to alternate platforms with ease.	Fully aware of the need for correctness of software and a strong believer in test-driven development. Extensively and comprehensively writes appropriate tests for methods before proceeding to integration phase
D	A strong grasp on object oriented principles and design principles with minor mistakes, including the capability to identify most of the issues issues with complex systems and the ability to solve the same.	Demonstrate a strong capability to identify requirements of a system, document and design the system using object oriented methodology, barring a few mistakes overall	Adequacy in using design patterns individually and in conjunction with others, based on requirements.	Capability to apply LO1 and LO3 in a technical environment. Extensive command over programming environments and platforms available. Needs some assistance to train with new systems.	Fully aware of the need for correctness of software and a strong believer in test-driven development. Adds unit tests before proceeding to integration phase but might be missing some test cases

CR	A significant understanding of object oriented principles and design principles with some gaps in knowledge, including the capability to identify partial issues with complex systems and the ability to solve the same	Demonstrate a average capability to identify requirements of a system, document and design the system using most of the concepts of object oriented methodology, but not all.	Adequacy in using design patterns individually, but not in conjunction with others, based on requirements.	Capability to apply LO1 and LO3 in a technical environment with some assistance. Familiarity with programming environments and platforms available. Cannot yet adapt to alternate platforms with ease.	Aware of the need for correctness of software and a accepts the need for test-driven development. Adds most of the unit tests before proceeding to integration phase but might be missing some test cases
P	A basic understanding of object oriented principles and design principles with significant gaps in knowledge, including the capability to identify partial issues with simple systems and the ability to solve the same	Demonstate an average capability to identify requirements of a familiar system, but needs assistance with documenting and designing the system.	Adequacy in applying design patterns individually, based on requirements in familiar scenarios but needs some assistance in applying design patterns in unchartered or unfamiliar situations.	Basic capability to apply LO1 and LO3 in a technical environment. Can use the basic features of programming environments and platforms available.	Performs test-driven development out of necessity rather than choice. Adds unit tests before proceeding to integration phase but might be misses crucial test cases

Grading

At the end of the semester, you will receive a grade that reflects your achievement in the unit

- **Fail (F)**: does not provide evidence of attainment of all learning outcomes. There is missing or partial or superficial or faulty understanding and application of the fundamental concepts in the field of study; and incomplete, confusing or lacking communication of ideas in ways that give little attention to the conventions of the discipline.
- **Pass (P)**: provides sufficient evidence of the achievement of learning outcomes. There is demonstration of understanding and application of fundamental concepts of the field of study; and communication of information and ideas adequately in terms of the conventions of the discipline. The learning attainment is considered satisfactory or adequate or competent or capable in relation to the specified outcomes.
- **Credit (Cr)**: provides evidence of learning that goes beyond replication of content knowledge or skills relevant to the learning outcomes. There is demonstration of substantial understanding of fundamental concepts in the field of study and the ability to apply these concepts in a variety of contexts; plus communication of ideas fluently and clearly in terms of the conventions of the discipline.
- **Distinction (D)**: provides evidence of integration and evaluation of critical ideas, principles and theories, distinctive insight and ability in applying relevant skills and concepts in relation to learning outcomes. There is demonstration of frequent originality in defining and analysing issues or problems and providing solutions; and the use of

means of communication appropriate to the discipline and the audience.

- **High Distinction (HD)**: provides consistent evidence of deep and critical understanding in relation to the learning outcomes. There is substantial originality and insight in identifying, generating and communicating competing arguments, perspectives or problem solving approaches; critical evaluation of problems, their solutions and their implications; creativity in application.

In this unit, your final grade depends on your performance in each part of the assessment. For each task, you receive a mark that combines your standard of performance regarding each learning outcome assessed by this task. Then the different component marks are added up to determine your total mark out of 100. Your grade then depends on this total mark and your overall standards of performance.

In particular, **in order to pass the unit**, you must

- Have performed satisfactorily in the internal (assessment) components of the course.
- Have satisfactory performance in the final examination.

This means that you may fail the unit if you do not submit satisfactory submissions for the assignments and do not perform satisfactorily in the exam.

Department of Computing expectations are that students have to perform satisfactorily in the final exam as well as in their internal work/assignments.

Obtaining a specific grade in this unit will require a student to obtain (in addition to the above):

- the required total number of marks (Pass - 50, Credit - 65, Distinction - 75, High Distinction - 85).
- Pass - a minimum of 40% (24/60) in the final exam
- Credit - a minimum of 60% (36/60) in the final exam
- Distinction - a minimum of 70% (42/60) in the final exam
- High Distinction - a minimum of 80% (48/60) in the final exam