ITEC629

Object-Oriented Programming Practices

S2 Day 2015

Dept of 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
Convenor
Annabelle McIver
annabelle.mciver@mq.edu.au
By appointment

Lecturer
Oldooz Dianat
oldooz.dianat@mq.edu.au
By appointment

Tutor
Matt Cabanag
matt.cabanag@mq.edu.au

Tutor
Kym Haines
kym.haines@mq.edu.au

Credit points
4

Prerequisites

Corequisites

Co-badged status
COMP229

Unit description
Object-oriented programming is a key technology for modern computing. This unit bridges the gap between introductory programming and larger multi-person projects by considering the use of object-oriented techniques to produce intermediate sized software. Practical exercises emphasise the importance of programming practices such as appropriate documentation, systematic approaches to debugging and testing, and the use of software development tools. The unit is taught using Java.

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:

Have an ability to use and apply application libraries in an OO programming language

Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments

Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.

Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns

Apply key OO concepts of concurrency in working code

Apply key OO concepts and design patterns to create and adapt libraries.

General Assessment Information

Standards

P

Can correctly reproduce basic "object oriented programming" facts and definitions. Can develop, test and document simple java programs that are similar to provided examples. Has demonstrated a working knowledge of GUI interfaces and a selection of common design patterns. Has demonstrated a working knowledge of applications involving concurrency.

CR

Can correctly reproduce basic "object oriented programming" facts and definitions, and can apply them in some unfamiliar contexts. Can develop, test and document simple java programs in situations similar to the provided examples and has demonstrated the ability to create non-trivial automated tests. Has demonstrated a working knowledge of GUI interfaces and a selection of common design patterns, and can apply them in unfamiliar situations. Has a sound understanding of concurrency, and is able to apply it to new designs, with the help of additional material which has bot been discussed.

D

Exhibits breadth and depth of understanding of object oriented concepts and issues and apply a variety of design techniques to develop object oriented code. Can use terminology accurately in new contexts. Can express ideas in their own words and has an understanding of the limits of their understanding. Is able to apply a wide range of documentation, debugging and testing practices in developing code. Can develop code with non-trivial GUI interfaces. Can construct implementations of common design patterns and apply them when writing code. Has a sound understanding of concurrency, and is able to apply it to new designs, with the help of additional material which has not been discussed.

HD

As for Cr or D and is aware of the context in which the concepts and issues are developed and their limitations. Able to generate and justify principles and hypotheses for existing or new concepts or issues. As for Cr or D, and is proficient in the application of OO libraries, and application of error handling. As for Cr or D and has well-developed skills for applying documentation, debugging and testing practices in ways that have not been previously illustrated by examples. As for Cr or D and can develop applications using techniques or approaches that have not been discussed. As for Cr or D and is aware of the context in which design pattern concepts are developed and their limitations. Able to generate and justify principles and hypotheses for existing or new design pattern concepts. Applies appropriate patterns consistently throughout the design and implementation of a complete software system, and can give a complete account of how those patterns interact to achieve a functionality objective. As for Cr or D and can develop applications using techniques or approaches that have not been discussed.

These assessment standards will be used to give a numeric mark to each assessment submission during marking. The mark will correspond to a letter grade for that task according to the University guidelines. The final raw mark for the unit will be calculated by combining the marks for all assessment tasks according to the given percentage weightings.

Completing the unit satisfactorily requires you to achieve a P grade (50% or more) for your raw mark. On occasion your raw mark for the unit may not be the same as the Standardised Numeric Grade (SNG) which you receive as the final result. Under the Senate guidelines, raw marks may be scaled to ensure that there is a degree of comparability across the university, so that units with the same past performances of their students should achieve similar results.

Assessment Tasks

Name	Weighting	Due
Weekly exercise	10%	Weeks 2--13
Diagnostic	5%	Week 4
Assignment 1	20%	Week 7
Assignment 2	20%	Week 12
Take Home Examination	45%	Week 13

Weekly exercise

Due: **Weeks 2--13**
Weighting: **10%**

In most weeks, you will be asked to submit a solution to a short programming exercise or problem set. No late submissions are accepted. Submissions are via iLearn.

On successful completion you will be able to:

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code

Diagnostic

Due: **Week 4**
Weighting: **5%**

In week 4 you will complete a selection of programming exercises, covering the basic material from weeks 1-3. This is intended to be an early diagnostic assessment. 20% late penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments

Assignment 1

Due: **Week 7**
Weighting: **20%**

This assignment consist of programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises or the early diagnostic test. 20% late penalty per day or part-day. Submissions are via iLearn.

ITEC students will be asked to complete an additional assessment task based on the COMP229

specification. The additional assessment task will be awarded up to 5% of the overall assignment mark.

On successful completion you will be able to:
- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts and design patterns to create and adapt libraries.

Assignment 2

Due: **Week 12**
Weighting: **20%**

This assignment consists of programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises or the early diagnostic test. 20% late penalty per day or part-day. Submissions are via iLearn.

ITEC students will be asked to complete an additional assessment task based on the COMP229 specification. The additional assessment task will be awarded up to 5% of the overall assignment mark.

On successful completion you will be able to:
- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code
- Apply key OO concepts and design patterns to create and adapt libraries.

Take Home Examination

Due: **Week 13**
Weighting: **45%**

This will be in the form of a take-home examination and you will be asked a range of long and short answer questions. You will be asked to submit the solutions via turnitin administered through iLearn. Please see the special instructions regarding disruption.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code

Delivery and Resources

Each week of ITEC629 has three hours of lectures and a one-hour tutorial class. The tutors will be available for half an hour consultation directly after the one hour class.

Close to the submission deadlines for the diagnostic, and assignments there will be an additional 1 hour practical class following directly after the 1 hour tutorial where tutors will be available with help and advice related to the assessment task. There will be one such additional hour for the diagnostic and 2 such additional hours for Assignments 1&2.

Classes commence in Week 2 and are held in the E6A 121 Computer Laboratory.

Required text

 Head First Design Patterns by Eric T Freeman, Elisabeth Robson, Bert Bates and Kathy Sierra, O'Rielly Media, October 2004 (ISBN:978-0-596-00712-6)

Note that this text book was introduced for the 2011 offering of COMP229, so you may be able to purchase a second hand copy from a former COMP229 student. Failing that, you will be able to purchase a copy from the University COOP Bookshop or from your favorite online purveyor of fine literary works.

You will find the lecture material much easier to understand if you read the text and lecture notes in advance of the lectures. Lecture notes (or slides), information about which chapters to read in the text book and any other relevant preparatory readings will be provided for you on the iLearn site (http://ilearn.mq.edu.au) for ITEC629 a couple days prior to each lecture.

Related texts

The Macquarie library contains many books on object-oriented programming in general, and on programming specifically in Java, that you may want to use to supplement the text and lecture notes.

One particularly useful service that the library provides you with is access to many Java related titles online via the Safari Books Online (http://proquest.safaribooksonline.com/) service. Using this service, which you can only access from a machine connected to the University network, you might like to have a look at the following Java titles:

1. Learning Java, 3rd Edition by Jonathan Knudsen; Patrick Niemeyer, ISBN: 978-0-596-00873-4

2. Java in a Nutshell, 5th Edition by David Flanagan, ISBN: 978-0-596-00773-7

3. Java Examples in a Nutshell, 3rd Edition by David Flanagan, ISBN: 978-0-596-00620-4

   The web itself is an ideal source of Java information, and from time to time we will be

   posting useful links on the COMP229 iLearn site. Two particularly useful resources are:

4. Thinking in Java by Bruce Eckel, a free version of the 3rd edition of this pretty

   comprehensive book is available for download from http://www.mindview.net/Books/TIJ/ and its 4th edition, which is updated for use with Java 5 and 6, is available from http://mindview.net/Books/TIJ4.

5. The official Java Tutorial http://download.oracle.com/javase/tutorial/ which is a comprehensive resource providing trails covering topics ranging from the basics of Java programming to more advanced subjects like GUI development, Generics, Class Reflection, Sound, Graphics, Network Programming and Concurrency

Unit webpage and technology

ITEC629 will make extensive use of the iLearn system for delivery of class materials, discussion boards, real time chat, submission of work and access to marks and feedback. Students should check the iLearn site (http://ilearn.mq.edu.au) regularly for unit updates.

Questions and general queries regarding the content of this unit, its tutorials or practicals should be posted to the appropriate discussion board on the ITEC629 iLearn site. In particular, any

questions which are of interest to all students in this unit should be posted to one of these discussion boards, so that everyone can benefit from the answers.

Echo360

Audio and screen video recordings of the lectures will be made available online via Echo360. A link to these recordings will be provided on the iLearn site for this unit

Technology Used and Required

The practical work in this unit involves programming in Java (www.java.com) using the Eclipse Integrated Development Environment (www.eclipse.org). We will also be using a distributed version control system called Mercurial (http://mercurial.selenic.com) to access shared code

repositories hosted on the BitBucket website (http://bitbucket.org).

This software is already installed for you in the 200 level computing labs, on the ground floor of building E6A.

Tools and libraries to support debugging, automated testing, GUI development and so forth will be introduced and used as the semester progresses. When that occurs you will be provided with full instructions in lectures on how to install and use each one.

Unit Schedule

Week	Topic
1	OO Concepts
2	Overloading/ Inheritance
3	Generics
4	Automated Testing/ JavaDoc
5	Observer, Decorator
6	Factory, Abstract Factory
7	Singleton, Adapter, Facade
8	Template, Iterator, Composite
9	Threads
10	Advanced Concurrency
11	GUI programming/ Input/Output
12	Even more design patterns (with applications)
13	Review; discussion

Assumed Knowledge

This unit requires:

• Experience writing programs in the Java programming language. In particular, we will assume that you are familar with all of the basic features of Java, including the use of strings, arrays, loops, control structures (such as if statements) and methods.

• Experience using the Eclipse IDE http://www.eclipse.org to develop simple Java applications.

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/

Results

Results shown in *iLearn*, or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit ask.mq.edu.au.

Special Consideration for the Take Home Examinations

These examinations are to test for students' comprehension of the taught material and it is very important that all students are compared fairly with their peers. Therefore for these assessments there will be no extensions and 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 for the entire period of the take home examination. In view of these strict conditions please make sure that you take time to read the assessment schedule set out in the unit outline, and mark the dates in your diary.

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

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Learning outcomes

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code
- Apply key OO concepts and design patterns to create and adapt libraries.

Assessment tasks

- Weekly exercise
- Diagnostic
- Assignment 1
- Assignment 2
- Take Home Examination

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

Learning outcomes

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code
- Apply key OO concepts and design patterns to create and adapt libraries.

Assessment tasks

- Weekly exercise
- Diagnostic
- Assignment 1
- Assignment 2
- Take Home Examination

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where

relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns
- Apply key OO concepts of concurrency in working code
- Apply key OO concepts and design patterns to create and adapt libraries.

Assessment tasks

- Weekly exercise
- Diagnostic
- Assignment 1
- Assignment 2
- Take Home Examination

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Have an ability to use and apply application libraries in an OO programming language
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- Understand the key concepts of object oriented programming, and apply them in practice

using an OO language to design and develop applications including GUI applications.

- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns

- Apply key OO concepts of concurrency in working code

- Apply key OO concepts and design patterns to create and adapt libraries.

Assessment tasks

- Weekly exercise

- Diagnostic

- Assignment 1

- Assignment 2

- Take Home Examination

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcomes

- Have an ability to use and apply application libraries in an OO programming language

- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments

- Understand the key concepts of object oriented programming, and apply them in practice using an OO language to design and develop applications including GUI applications.

- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns

- Apply key OO concepts of concurrency in working code

- Apply key OO concepts and design patterns to create and adapt libraries.

Assessment tasks

- Weekly exercise

- Diagnostic

- Assignment 1

- Assignment 2

- Take Home Examination

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Assessment tasks

- Weekly exercise
- Diagnostic
- Assignment 1
- Assignment 2
- Take Home Examination