COMP229

Object-Oriented Programming Practices

S2 Day 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
Matthew Roberts
matthew.roberts@mq.edu.au
Contact via matthew.roberts@mq.edu.au
E6A 374
Monday 11-12, Friday 11-12, or by appointment

Lecturer
Luke Mathieson
luke.mathieson@mq.edu.au
Contact via luke.mathieson@mq.edu.au
Monday 1-2

Christophe Doche
christophe.doche@mq.edu.au

Credit points
3

Prerequisites
COMP125(P) or COMP165(P)

Corequisites

Co-badged status

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:

Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

Have an ability to use application libraries, in this case the Java SE API

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

Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)

Apply key OO concepts and libraries to design and develop GUI applications

Apply key OO concepts of concurrency in working code.

Assessment Tasks

Name	Weighting	Due
Weekly Exercises	8%	Weeks 1-12
Assignment One	6%	Week 4
Module Examinations	44%	Week 5, Week 9, Week 12
Assignment Two	14%	Week 8
Assignment Three	20%	Week 13
The Reading Game	8%	ongoing

Weekly Exercises

Due: **Weeks 1-12**
Weighting: **8%**

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:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop GUI applications
- Apply key OO concepts of concurrency in working code.

Assignment One

Due: **Week 4**
Weighting: **6%**

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:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)

Module Examinations

Due: **Week 5, Week 9, Week 12**
Weighting: **44%**

On three occasions during semester and once in the final exam period students sit a module examination. The three in-semester exams are run in lab classes and each offers a different exam. In week 5 students will sit the Java Programming Practice exam, in week 9 students will sit the Design Patterns exam, and in week 12 students will sit the Concurrency exam. All three exams are then offered a second time in the exam period (with different questions of course).

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API

- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop GUI applications
- Apply key OO concepts of concurrency in working code.

Assignment Two

Due: **Week 8**
Weighting: **14%**

The assignments are 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 penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)

Assignment Three

Due: **Week 13**
Weighting: **20%**

The assignments are 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 penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Apply key OO concepts and libraries to design and develop GUI applications
- Apply key OO concepts of concurrency in working code.

The Reading Game

Due: **ongoing**
Weighting: **8%**

The Reading Game is a question and answer game built by a class of participants, to create a collective learning space where every action serves to introduce, build, or clarify concepts from the course material. The quality of the questions and the contents of the quizzes are up to the participants who receive points for their efforts in both asking and answering questions. The game leverages game mechanics to make the participants' interactions with the game, fun. Participants can rate questions, which directly impact the contents of review quizzes, while also activating a secondary reward called 'stars' for participants whose questions are deemed outstanding by their peers. Participants can progress on to the next level of asking open questions directly to the teachers and their cohort, by using their accumulated points.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)

Delivery and Resources

CLASSES

Each week of COMP229 has three hours of lectures and a two-hour mixed class (i.e., two hour combined tutorial and practical). The Mixed classes commence in Week 2 and are held in the E6A 121 Computer Laboratory.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

The required text for the unit is:

- 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 COMP229 a couple days prior to each lecture.

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 iLeran 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 for $25 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 USED AND REQUIRED

Online Resources

COMP229 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 COMP229 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	Module	Topic	Module Exam	Assignment
1	Java Programming Practice	OO Concepts		
2	Java Programming Practice	Overloading/Inheritance		
3	Java Programming Practice	Generics		
4	Java Programming Practice	Automated Testing/JavaDoc		Submit Assignment One
5	Design Patterns	Observer, Decorator	Java Programming Practice	
6	Design Patterns	Factory, Abstract Factory		
7	Design Patterns	Singleton, Adapter, Facade		
		Mid-Semester Break		
8	Design Patterns	Template, Iterator, Composite		Submit Assignment Two
9	Concurrency	Threads	Design Patterns	
10	Concurrency	Advanced Concurrency		
11	Concurrency	GUI programming/Input/Output		

12		Even more design patterns (with applications)	Concurrency	
13		Review; discussion		Submit Assignment Three
Exam Period			Java Programming Practice, Design Patterns, Concurrency	

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/

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.

Plagiarism and Academic Honesty

Please refer to the Macquarie University academic honesty policy to find out more about what is expected from you with regard to issues such as plagiarism, deception, fabrication and sabotage.

In particular, you should consult the schedule of penalties which may apply to infringements of this academic honesty policy.

You should be aware that we use a range of powerful source code analysis tools to detect cases of plagiarism amongst assignment and weekly exercise submissions. Furthermore, we also use the Turnitin system to detect plagiarism in take home exams. So if you plagiarise the work of others it is likely that we will catch you out and apply the strongest penalty which applies under the University's academic honesty policy.

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

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

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop GUI applications

Assessment tasks

- Weekly Exercises
- Assignment One
- Module Examinations
- Assignment Two
- Assignment Three
- The Reading Game

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

- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade,

decorator etc)

Assessment tasks

- Weekly Exercises
- Assignment One
- Module Examinations
- Assignment Two
- Assignment Three
- The Reading Game

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 outcome

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

Assessment tasks

- Weekly Exercises
- Assignment One
- Module Examinations
- Assignment Two
- Assignment Three
- The Reading Game

Engaged and Ethical Local and Global citizens

As local citizens our graduates will be aware of indigenous perspectives and of the nation's historical context. They will be engaged with the challenges of contemporary society and with knowledge and ideas. We want our graduates to have respect for diversity, to be open-minded, sensitive to others and inclusive, and to be open to other cultures and perspectives: they should have a level of cultural literacy. Our graduates should be aware of disadvantage and social justice, and be willing to participate to help create a wiser and better society.

This graduate capability is supported by:

Learning outcome

- Apply key OO concepts of concurrency in working code.

Assessment tasks

- Weekly Exercises
- Module Examinations
- Assignment Three

Assessment Standards

COMP229 will be graded according to the following general descriptions of the letter grades as specified by Macquarie University.

- **High Distinction (HD, 85-100)**: 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.

- **Distinction (D, 75-84)**: 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.

- **Credit (Cr, 65-74)**: 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.

- **Pass (P, 50-64)**: 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.

- **Fail (F, 0-49)**: 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.

Please note that, since the beginning 2011, Macquarie University no longer awards Conceded Pass (PC) grades in its units.

The standards of achievement that will be used to assess each of the assessment tasks with respect to the letter grades are as follows. Where applicable, more specific versions of these standards will be provided with the assessment task descriptions.

Learning Outcome 1: Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

P	Can correctly reproduce basic facts and definitions across a breadth of concepts and issues, but lacks depth of understanding.
Cr or D	Exhibits breadth and depth of understanding of concepts and issues. Can use terminology accurately in new contexts. Can express ideas in their own words and has an understanding of the limits of their understanding.
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.

Learning Outcome 2: Have an ability to use application libraries, in this case the Java SE APII;

P	Can develop java programs that are similar to provided examples, and understands how to create basic tests.
Cr or D	Can create and understand designs that differ from provided examples and apply a variety of design techniques to their solution.
HD	As for Cr or D, and is proficient in the application of OO libraries, and application of error handling.

Learning Outcome 3: Be familiar with good programming practices such as testing, debugging, documentation and version control;

P	Can apply some basic documentation, debugging and testing practices along the lines of examples provided.
Cr or D	Is able to apply a wide range of documentation, debugging and testing practices to their code along the lines of examples provided. Has the ability to create non-trivial automated tests.
HD	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.

Learning Outcome 5: Ability to apply key OO concepts and libraries to develop GUI applications;

P	Can put together code samples to create a non-trivial application involving a full GUI interface, with some implementation of known patterns.
Cr or D	As for P and has a sound understanding of a number GUI concepts, and is able to apply them to new designs, with the help of additional material which has bot been discussed.
HD	As for Cr or D and can develop applications using techniques or approaches that have not been discussed.

Learning Outcome 4: Understanding of the concepts underlying design patterns and a working knowledge of a number of well known design patterns (singleton, observer, factory, iterator, decorator etc);

P	Can correctly reproduce basic facts and definitions relating to design patterns and can identify opportunities for their application, but lacks depth of understanding.
Cr or D	Exhibits breadth and depth of understanding of design pattern concepts and issues. Can use related terminology accurately in new contexts. Can construct implementations of common design patterns and apply them when writing code. Can express these ideas in their own words and has an understanding of the limits of their understanding.
HD	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.

Learning Outcome 6: Ability to apply key OO concepts and libraries to concurrent applications;

P	Can put together code samples to create a non-trivial application involving concurrency.
Cr or D	As for P and 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.
HD	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 percentage weightings shown in the assessment summary above.

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.

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.

Changes since Last Offering

We have moved from take-home exams to in-class module exams.