

# **COMP2000**

# **Object-Oriented Programming Practices**

Session 2, Special circumstance 2020

Department of Computing

# Contents

General Information	2
Learning Outcomes	2
General Assessment Information	3
Assessment Tasks	3
Delivery and Resources	6
Unit Schedule	8
Policies and Procedures	9

#### 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.

#### Notice

As part of Phase 3 of our return to campus plan, most units will now run tutorials, seminars and ot her small group learning activities on campus for the second half-year, while keeping an online ver sion available for those students unable to return or those who choose to continue their studies onli ne.

To check the availability of face-to-face and onlin e activities for your unit, please go to timetable vi ewer. To check detailed information on unit asses sments visit your unit's iLearn space or consult yo ur unit convenor.

## **General Information**

Unit convenor and teaching staff Damian Jurd damian.jurd@mq.edu.au

Matthew Roberts matthew.roberts@mq.edu.au

Credit points 10

Prerequisites COMP1010 or COMP125

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 <a href="https://www.mq.edu.au/study/calendar-of-dates">https://www.mq.edu.au/study/calendar-of-dates</a>

# Learning Outcomes

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

**ULO1:** explain the key concepts of object oriented programming, and program proficiently in an OO programming language

**ULO2:** apply the concepts underlying software design and a working knowledge of a selection of well known design patterns

**ULO3:** demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments

**ULO4:** apply key object oriented concepts and libraries to design and develop applications of significant complexity

ULO5: apply key concepts of concurrency theoretically and in working code

# **General Assessment Information**

## LATE PENALTY

No extensions will be granted without an approved application for <u>Special Consideration</u>. There will be a deduction of 20% of the total available marks made from the total awarded mark for each 24 hour period or part thereof that the submission is late. For example, 25 hours late in submission for an assignment worth 40 marks – 40% penalty – deducted from the total for an assignment marked out of 100. No submission will be accepted after solutions have been posted.

## EXAMINATIONS

The unit has three module examinations which assess the material studied in weeks 1-4, 5-9, and 10-12, respectively. Each examination is offered twice: once during the workshop classes in weeks 5, 10 and 13, respectively, and once in the final examination period. The repeat offerings of the examination will not be identical examinations but will be designed to assess the same material. All examinations delivered electronically via iLearn in the computing laboratories on level 1 of building 9WW.

If a student attempts an examination more than once then the higher of their marks for the two attempts will be used to compute the grade.

If you receive <u>Special Consideration</u> for the final exam, a supplementary exam will be scheduled after the normal exam period, following the release of marks. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

# Assessment Tasks

Name	Weighting	Hurdle	Due
Weekly Tasks	16%	No	Weeks 1-4, 6-9, 11, 12
Module Examinations	44%	No	Weeks 5, 10, 13, Exam Period
Assignment One	8%	No	Week 4
Assignment Two	14%	No	Week 8

Name	Weighting	Hurdle	Due
Assignment Three	18%	No	Week 12

### Weekly Tasks

Assessment Type <sup>1</sup>: Participatory task Indicative Time on Task <sup>2</sup>: 12 hours Due: **Weeks 1-4, 6-9, 11, 12** Weighting: **16%** 

Each week you will be set a task to complete which will be submitted online or checked in a workshop session.

On successful completion you will be able to:

- explain the key concepts of object oriented programming, and program proficiently in an OO programming language
- apply the concepts underlying software design and a working knowledge of a selection of well known design patterns
- demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- apply key object oriented concepts and libraries to design and develop applications of significant complexity
- apply key concepts of concurrency theoretically and in working code

### Module Examinations

Assessment Type 1: Examination Indicative Time on Task 2: 24 hours Due: **Weeks 5, 10, 13, Exam Period** 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 practical workshops and each offers a different exam.

On successful completion you will be able to:

- explain the key concepts of object oriented programming, and program proficiently in an OO programming language
- apply the concepts underlying software design and a working knowledge of a selection of well known design patterns
- demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- apply key object oriented concepts and libraries to design and develop applications of significant complexity
- apply key concepts of concurrency theoretically and in working code

## Assignment One

Assessment Type 1: Programming Task Indicative Time on Task 2: 5 hours Due: **Week 4** Weighting: **8%** 

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises.

On successful completion you will be able to:

- explain the key concepts of object oriented programming, and program proficiently in an OO programming language
- demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments

### Assignment Two

Assessment Type 1: Programming Task Indicative Time on Task 2: 10 hours 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.

On successful completion you will be able to:

- explain the key concepts of object oriented programming, and program proficiently in an OO programming language
- apply the concepts underlying software design and a working knowledge of a selection of well known design patterns
- demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- apply key object oriented concepts and libraries to design and develop applications of significant complexity

## Assignment Three

Assessment Type <sup>1</sup>: Programming Task Indicative Time on Task <sup>2</sup>: 12 hours Due: **Week 12** Weighting: **18%** 

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises.

On successful completion you will be able to:

- demonstrate good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments
- apply key concepts of concurrency theoretically and in working code

<sup>1</sup> If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

<sup>2</sup> Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation

# **Delivery and Resources**

# CLASSES

Each week of COMP2000 has one hour of lectures and a two-hour mixed class (i.e., two hour combined tutorial and practical). The Mixed classes commence in Week 1 and are held in the Computer Laboratory in room 9WW 121 (Wednesdays and Thursdays) or 9WW 123 (Fridays). In

all cases students are expected to do significant preparatory work, readings and exercises *before* attending classes.

# 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 the unit (then known as 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 favourite online purveyor of fine literary works.

There will be no lecture notes provided, all examinable material is given in course readings and the textbook. Students are required to study this material and answer preparatory questions *before* class.

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:

- 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
- 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 for \$25 from http://mindview.net/Books/TIJ4.
- 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**

COMP2000 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 COMP2000 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 (<u>https://www.java.com</u>) using the Microsoft Visual Studio Code IDE (<u>https://code.visualstudio.com</u>). We will also be using a distributed version control system called Git to access shared code repositories hosted on the BitBucket website (<u>https://bitbucket.org</u>).

This software is already installed for you in the 200 level computing labs (ground floor, 9 Wally's Walk) and it is available to download, install and use for free on your own machine(s). It should work equally well on Mac OSX, Linux or Microsoft Windows platforms.

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**

<u>week</u>	lecture	lecturer	workshop	<u>submit</u>
1	Version Control	DJ	Version Control	
2	Java Classes and Objects	DJ	Java Classes and Objects	
3	Inheritance and Overloading	DJ	Inheritance and Overloading	
4	Generics	DJ	Generics	assign 1
5	Exceptions, Introduction to Patterns	DJ	Exam: Java Concepts	
6	Strategy Pattern, Observer Pattern	DJ	Observer Pattern	
7	Decorator Pattern, Iterator/Composite Patterns	MR	Decorator Pattern, Iterator/Composite Patterns	
8	Singleton Pattern, Command Pattern	MR	Singleton Pattern, Command Pattern	assign 2
9	Concurrency	MR	Concurrency	

10	Concurrency	MR	Exam: Design Patterns	
11	Concurrency	MR	Concurrency	
12	Concurrency	MR	Concurrency	assign 3
13	Review	MR	Exam: Concurrency	

# **Policies and Procedures**

Macquarie University policies and procedures are accessible from Policy Central (https://staff.m q.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-centr al). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy
- Fitness to Practice Procedure
- Grade Appeal Policy
- Complaint Management Procedure for Students and Members of the Public
- <u>Special Consideration Policy</u> (*Note: The Special Consideration Policy is effective from 4* December 2017 and replaces the Disruption to Studies Policy.)

Students seeking more policy resources can visit the <u>Student Policy Gateway</u> (https://students.m <u>q.edu.au/support/study/student-policy-gateway</u>). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

If you would like to see all the policies relevant to Learning and Teaching visit Policy Central (http s://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/p olicy-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/study/getting-started/student-conduct

### Results

Results published on platform other than <u>eStudent</u>, (eg. iLearn, Coursera etc.) 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 <u>eStudent</u>. For more information visit <u>ask.mq.edu.au</u> or if you are a Global MBA student contact globalmba.support@mq.edu.au

### Student Support

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

### dents.mq.edu.au/support/

### **Learning Skills**

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

- · Getting help with your assignment
- Workshops
- StudyWise
- Academic Integrity Module

The Library provides online and face to face support to help you find and use relevant information resources.

- Subject and Research Guides
- Ask a Librarian

### 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

If you are a Global MBA student contact globalmba.support@mq.edu.au

### IT Help

For help with University computer systems and technology, visit <u>http://www.mq.edu.au/about\_us/</u>offices\_and\_units/information\_technology/help/.

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