



COMP2000

Object-Oriented Programming Practices

Session 2, In person-scheduled-weekday, North Ryde 2026

School of Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>General Assessment Information</u>	3
<u>Assessment Tasks</u>	4
<u>Delivery and Resources</u>	6
<u>Unit Schedule</u>	6
<u>Policies and Procedures</u>	7
<u>Changes from Previous Offering</u>	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.

General Information

Unit convenor and teaching staff

Matthew Roberts

matthew.roberts@mq.edu.au

Credit points

10

Prerequisites

COMP1010

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:

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

Requirements to Pass this Unit

To pass this unit you must achieve a total mark equal to or greater than 50%. There are no hurdle requirements for the unit.

LATE ASSESSMENT SUBMISSION PENALTY

Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark of the task) will be applied for each day a written report or presentation assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is 11:55 pm. A 1-hour grace period will be provided to students who experience a technical concern. For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration. For example, if the assignment is worth 25 marks (of the entire unit) and your submission is late by 19 hours (or 23 hours 59 minutes 59 seconds), 1.25 marks (5% of 25 marks) will be deducted. If your submission is late by 24 hours (or 47 hours 59 minutes 59 seconds), 2.5 marks (10% of 25 marks) will be deducted, and so on.

ASSESSMENTS WHERE LATE SUBMISSIONS WILL BE ACCEPTED

• Assignment 1: YES, Standard Late Penalty applies • Assignment 2: YES, Standard Late Penalty applies • Exam: NO

Special Consideration

The Special Consideration Policy aims to support students who have been impacted by shortterm circumstances or events that are serious, unavoidable and significantly disruptive, and which may affect their performance in assessment.

If you experience circumstances or events that affect your ability to complete either of the assignments in this unit on time, please inform the convenor and submit a Special Consideration request through ask.mq.edu.au.

AI Approach

- **Open** These assessments do not restrict the application of AI tools. However, it is expected that the use of AI is acknowledged in an appropriate manner, similar to the expectation to provide references for other resources or tools that are used in producing your work. Any use of AI should comply with the University's [Responsible and Ethical Use of Artificial Intelligence Policy](#).
- **Observed** These tasks are either fully or partly observed or invigilated. AI use will either

not be permitted or will be restricted during the assessment.

For more information, please see <https://students.mq.edu.au/study/assessment-exams/assessments/ai-and-assessment>

Assessment Tasks

Name	Weighting	Hurdle	Due	Groupwork/Individual	Short Extension	AI Approach
Final Exam	40%	No	Exam Period	Individual	No	Observed
Assignment 1	30%	No	13/09/2026	Individual	Yes	Open
Assignment 2	30%	No	08/11/2026	Individual	Yes	Open

Final Exam

Assessment Type ¹: Examination

Indicative Time on Task ²: 16 hours

Due: **Exam Period**

Weighting: **40%**

Groupwork/Individual: Individual

Short extension ³: No

AI Approach: Observed

You will undertake a final examination during the formal examination period.

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 1

Assessment Type ¹: Experiential task

Indicative Time on Task ²: 16 hours

Due: **13/09/2026**

Weighting: **30%**

Groupwork/Individual: Individual

Short extension ³: Yes

AI Approach: Open

A programming task that assesses your ability to put Object Oriented Programming theory into practice

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 2

Assessment Type ¹: Experiential task

Indicative Time on Task ²: 20 hours

Due: **08/11/2026**

Weighting: **30%**

Groupwork/Individual: Individual

Short extension ³: Yes

AI Approach: Open

A programming task that assesses your ability to put the second half of the course material into practical use.

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

¹ 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
- [Academic Success](#) for academic skills support.

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

³ An automatic short extension is available for some assessments. Apply through the [Service Connect Portal](#).

Delivery and Resources

COMP2000 is taught via weekly 2-hour lectures and practical classes. Each week you should participate in your scheduled two hour practical. For details of scheduled classes consult the timetables webpage.

Week1

Note that classes commence in week 1. The week-by-week details of the practical classes Unit guide COMP2000 will be available from iLearn.

Students are expected to do significant preparatory work, readings and exercises before attending classes.

Textbook and Reading Materials

The required texts for the unit are Learning Java 3rd Ed, Head First Design Patterns 2nd Ed, and Modern Java in Action 2nd Ed. All three are freely available online.

Each week a set of notes, a Rapid Aptitude Test (RAT), an application exercise, and self-study material will be provided in a pdf file. This material will include a set of activities which are to be done before class.

Communication Methods in COMP2000

We will communicate with you via your university email and through announcements on iLearn. Queries to convenors can either be placed on the iLearn discussion board or sent to the unit convenor via the contact email on iLearn.

Unit Schedule

Week 1 — Team Based Learning Team formation, team contracts, course logistics. No preparation required.

Week 2 — Git Distributed version control: init, clone, add, commit, log, push, pull. Preparation: Read Atlassian Git tutorial pages. Attend the lecture in week 1.

Week 3 — Objects and Classes Classes, methods, object creation, Java API documentation. Preparation: Read Learning Java Ch 5. Attend the lecture in week 2.

Week 4 — Inheritance and Overloading Subclassing, interfaces, inner classes, method overloading. Preparation: Read Learning Java Ch 6. Attend the lecture in week 3.

Week 5 — Generics Type erasure, wildcards, bounded types, generic methods. Preparation: Read Learning Java Ch 8. Attend the lecture in week 4.

Week 6 — Exceptions Try/catch/finally, checked vs unchecked exceptions, call stack. Preparation: Read Learning Java Ch 4.5. Attend the lecture in week 5.

Week 7 — Patterns 1: Strategy and Observer Design patterns: Strategy, Observer. Mid-semester submission due. Preparation: Read Head First Design Patterns Ch 1–2. Attend the lecture in week 6.

Week 8 — Patterns 2: Decorator, Iterator, State Design patterns: Decorator, Iterator, State. Preparation: Read Head First Design Patterns Ch 3, 9, 10. Attend the lecture in week 7.

Week 9 — Behaviour Parameterisation Lambdas, functional interfaces, method references. Preparation: Read Modern Java in Action Ch 1–2. Attend the lecture in week 8.

Week 10 — Streams Stream pipelines, intermediate/terminal operations, filter/map/reduce. Preparation: Read Modern Java in Action Ch 4–5. Attend the lecture in week 9.

Week 11 — Collecting Collectors, groupingBy, partitioningBy, custom collectors. Preparation: Read Modern Java in Action Ch 6. Attend the lecture in week 10.

Week 12 — Parallelism Threads, parallel streams, synchronisation, race conditions. Preparation: Read Modern Java in Action Ch 7. Attend the lecture in week 11.

Week 13 — Student Showcase Conference-style presentations. Final submission due. Preparation: Prepare a short presentation. Attend the lecture in week 12.

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central \(https://policies.s.mq.edu.au\)](https://policies.s.mq.edu.au). 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](#)
- [Assessment Procedure](#)
- [Complaints Resolution Procedure for Students and Members of the Public](#)
- [Special Consideration Policy](#)

Students seeking more policy resources can visit [Student Policies \(https://students.mq.edu.au/support/study/policies\)](https://students.mq.edu.au/support/study/policies). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

To find other policies relating to Teaching and Learning, visit [Policy Central \(https://policies.mq.edu.au\)](https://policies.mq.edu.au) and use the [search tool](#).

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: <https://students.mq.edu.au/admin/other-resources/student-conduct>

Results

Results published on platform other than [eStudent](#), (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 [eStudent](#). For more information visit connect.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

Academic Integrity

At Macquarie, we believe [academic integrity](#) – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a range of resources and services to help you reach your potential, including free [online writing and maths support](#), [academic skills development](#) and [wellbeing consultations](#).

Student Support

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

Academic Success

[Academic Success](#) provides resources to develop your English language proficiency, academic writing, and communication skills.

- [Ask a Study Coach](#)
- [Access StudyWISE](#)
- [Upload an assignment to Studiosity](#)
- [Complete the 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

Macquarie University offers a range of [Student Support Services](#) including:

- [IT Support](#)

- [Accessibility and disability support](#) with study
- Mental health [support](#)
- [Safety support](#) to respond to bullying, harassment, sexual harassment and sexual assault
- [Social support including information about finances, tenancy and legal issues](#)
- [Student Advocacy](#) provides independent advice on MQ policies, procedures, and processes

Student Enquiries

Got a question? Ask us via the [Service Connect Portal](#), or contact [Service Connect](#).

IT Help

For help with University computer systems and technology, visit <https://students.mq.edu.au/support/technology/service-desk>.

When using the University's IT, you must adhere to the [Acceptable Use of IT Resources Policy](#). The policy applies to all who connect to the MQ network including students.

Changes from Previous Offering

Fuller adoption of EPIC educational principals.

Unit information based on version 2026.03 of the [Handbook](#)