



## UNSW Course Outline

# COMP2511 Object-Oriented Design & Programming - 2024

Published on the 05 Feb 2024

## General Course Information

Course Code : COMP2511

Year : 2024

Term : Term 1

Teaching Period : T1

Is a multi-term course? : No

Faculty : Faculty of Engineering

Academic Unit : School of Computer Science and Engineering

Delivery Mode : Multimodal

Delivery Format : Standard

Delivery Location : Kensington

Campus : Sydney

Study Level : Undergraduate

Units of Credit : 6

### Useful Links

[Handbook Class Timetable](#)

## Course Details & Outcomes

### Course Description

COMP2511 builds on the foundation of first-year CSE courses, and begins to answer the questions: "What does good software look like?" and "How do you build software that is flexible, reusable and maintainable". The course introduces students to the Object-Oriented Programming

paradigm and explores how OOP attempts to solve the problem of good software design. Students explore fundamental Design Patterns in designing, writing and testing software, and how to apply these strategies to both theoretical and real-world problems. The course also teaches an appreciation for elegantly written code, problem solving and finding well-designed solutions to problems that have longevity of software in mind.

## Course Aims

This course aims to introduce students to the foundational principles of designing software that is flexible, reusable and maintainable. Students are exposed to the Object-Oriented Programming paradigm and explore how OOP attempts to solve the problem of good software design, while learning of other paradigms such as the functional paradigm. Students explore fundamental Design Patterns in writing and testing of software and their role in architecture of software, applying these solutions to both theoretical and real-world problems. The course teaches an appreciation for elegantly written code, problem solving and finding well-designed solutions to problems that have longevity of software systems in mind.

The course comes at a mid-point in the program, after foundational programming courses and before applying design principles to build large software systems.

# Course Learning Outcomes

Course Learning Outcomes
CL01 : Develop an appreciation for elegantly written software, and how to create and maintain well-designed systems
CL02 : Understand different design paradigms and methodologies, their background and application
CL03 : Apply the principles of Object-Oriented Design to solve problems
CL04 : Apply widely used Design Patterns to create extensible designs
CL05 : Create medium-scale systems from scratch, and work on existing systems as part of the Software Development Life Cycle
CL06 : Work with an enterprise programming language and IDE
CL07 : Explore the human factors at play in Software Design and Engineering; including ambiguity, the unknown, and trade-offs

Course Learning Outcomes	Assessment Item
CL01 : Develop an appreciation for elegantly written software, and how to create and maintain well-designed systems	<ul style="list-style-type: none"> <li>• Assignment 2 - Refactoring &amp; Design Patterns</li> </ul>
CL02 : Understand different design paradigms and methodologies, their background and application	<ul style="list-style-type: none"> <li>• Coursework</li> <li>• Final Exam</li> <li>• Assignment 2 - Refactoring &amp; Design Patterns</li> </ul>
CL03 : Apply the principles of Object-Oriented Design to solve problems	<ul style="list-style-type: none"> <li>• Assignment 1 - Object-Oriented Design</li> <li>• Coursework</li> <li>• Final Exam</li> <li>• Assignment 2 - Refactoring &amp; Design Patterns</li> </ul>
CL04 : Apply widely used Design Patterns to create extensible designs	<ul style="list-style-type: none"> <li>• Coursework</li> <li>• Final Exam</li> <li>• Assignment 2 - Refactoring &amp; Design Patterns</li> </ul>
CL05 : Create medium-scale systems from scratch, and work on existing systems as part of the Software Development Life Cycle	<ul style="list-style-type: none"> <li>• Assignment 1 - Object-Oriented Design</li> <li>• Assignment 2 - Refactoring &amp; Design Patterns</li> </ul>
CL06 : Work with an enterprise programming language and IDE	<ul style="list-style-type: none"> <li>• Assignment 1 - Object-Oriented Design</li> <li>• Coursework</li> <li>• Final Exam</li> </ul>
CL07 : Explore the human factors at play in Software Design and Engineering; including ambiguity, the unknown, and trade-offs	<ul style="list-style-type: none"> <li>• Coursework</li> </ul>

# Learning and Teaching Technologies

Moodle - Learning Management System | Microsoft Teams | Blackboard Collaborate | Echo 360 | EdStem | CSE labs, Webcams, Gitlab, Git

## Learning and Teaching in this course

### Technology Stack

- [GitLab + Git](#) for version control;
- [WebCMS](#) for course notices and formal course material.
- **Edstem** for the course forum;
- **VSCode IDE** . While students are welcome to use any IDE for their work, the course will only be able to provide support for VSCode. **We highly recommend all students setup their IDE to work on their local machine** . VSCode via SSH/SSHFS cause problems with VSCode extensions.
- **Java Version 11** for development. Instructions on setting up Java are provided in the Week 1 lab.
- **Gradle Version 7.2** for dependency and build management. Instructions on setting up Gradle are provided in the Week 1 lab.
- **MS Teams** for hybrid lectures and online tutorials.

## Additional Course Information

### Laboratory Classes

Following the tutorial class each week, there will be a two-hour laboratory class, during which you will work on a variety of practical exercises that will enable you to apply the concepts taught in the lectures.

Each week, there will be several exercises to work on. The exercises will need to be submitted and will be assessed by your tutor. You will need to demonstrate your work to your tutor during the following week's lab. In some lab exercises, you may need to complete a reflective blog post outlining your thought process, approach and learnings.

In the later weeks of the term, the laboratory sessions will also be used to conduct oral assessments on assignment submissions.

Your coursework marks will be based on your lab marks; read the Coursework (15%) section below for further information.

As this course has a significant practical component, the weekly labs are important to help you acquire the necessary skills in the relevant tools and frameworks that will be used in the

implementation of the assignments and also prepare you for the Final Exam. If you do not put a good amount of effort into the weekly labs you risk failing the Assignments and the Final Exam.

Solutions are not released for lab exercises due to plagiarism and educational reasons. Instead, more complex labs will have a retrospective 'walkthrough' video released.

There are no late extensions on lab exercises, since we release the retrospectives directly following when the labs are due. Students who have extenuating circumstances and cannot complete their lab on time will need to email [cs2511@cse.unsw.edu.au](mailto:cs2511@cse.unsw.edu.au) and we will take an average of your other weeks' labs for your lab mark for that week.

# Assessments

## Assessment Structure

Assessment Item	Weight	Relevant Dates
Coursework Assessment Format: Individual	15%	
Assignment 2 - Refactoring & Design Patterns Assessment Format: Group	20%	
Assignment 1 - Object-Oriented Design Assessment Format: Individual	15%	
Final Exam Assessment Format: Individual	50%	

## Assessment Details

### Coursework

#### Assessment Overview

Each week there will be a series of core activities, consisting of a series of Design & Programming lab exercises.

In addition to these core activities, students will have the opportunity to undertake choice activities and provide further engagement via the course forum.

The tutor will assess each student's work and provide feedback on it.

#### Course Learning Outcomes

- CL02 : Understand different design paradigms and methodologies, their background and application
- CL03 : Apply the principles of Object-Oriented Design to solve problems

- CL04 : Apply widely used Design Patterns to create extensible designs
- CL06 : Work with an enterprise programming language and IDE
- CL07 : Explore the human factors at play in Software Design and Engineering; including ambiguity, the unknown, and trade-offs

## **Assignment 2 - Refactoring & Design Patterns**

### **Assessment Overview**

This assignment is completed in pairs, where both members come from the same tutorial class. In classes where there are odd numbers, the tutor will form one group of three.

The assignment requires students to take an existing piece of software and refactor it so that it can more easily adapt to changes in the specification. The specification will change during the course of this assignment.

Both members of the pair must contribute to the completion of Assignment II; pairs with unequal contribution will have their mark individually adjusted. Students are expected to communicate proactively (while the assignment is being worked on) with their tutor regarding issues in contribution, rather than after the assignment is complete.

The marking criteria for the assignment is given in the assignment specification, and the tutors will mark the submissions using these criteria and provide feedback.

### **Course Learning Outcomes**

- CL01 : Develop an appreciation for elegantly written software, and how to create and maintain well-designed systems
- CL02 : Understand different design paradigms and methodologies, their background and application
- CL03 : Apply the principles of Object-Oriented Design to solve problems
- CL04 : Apply widely used Design Patterns to create extensible designs
- CL05 : Create medium-scale systems from scratch, and work on existing systems as part of the Software Development Life Cycle

## **Assignment 1 - Object-Oriented Design**

### **Assessment Overview**

This assignment involves carrying out object-oriented design on a non-trivial problem, and implementing this design in Java.

The marking criteria for the assignment is given in the assignment specification, and the tutors will mark the submissions using these criteria and provide feedback.

### Course Learning Outcomes

- CL03 : Apply the principles of Object-Oriented Design to solve problems
- CL05 : Create medium-scale systems from scratch, and work on existing systems as part of the Software Development Life Cycle
- CL06 : Work with an enterprise programming language and IDE

## Final Exam

### Assessment Overview

There will be a 3-hour in person (face-to-face) final exam in the CSE labs during the UNSW exam period.

The exam will consist of a mix of programming , design and analysis questions.

There is no hurdle requirement for the exam.

### Course Learning Outcomes

- CL02 : Understand different design paradigms and methodologies, their background and application
- CL03 : Apply the principles of Object-Oriented Design to solve problems
- CL04 : Apply widely used Design Patterns to create extensible designs
- CL06 : Work with an enterprise programming language and IDE

## General Assessment Information

### Assignments

- **Assignment I** will be completed **individually** , released at the beginning of Week 2 and due at the end of Week 5.
- **Assignment II** will be completed **in pairs** , released in the middle of Week 5 and is due at the end of Week 9.
- **All pairs must be from the same tutorial / lab** . In classes where there are odd numbers, the tutor will form one group of three. Pairs for Assignment II will be formed in Week 3 and finalised in Week 4.
- Both members of the pair must contribute to the completion of Assignment II; pairs with unequal contribution will have their mark individually adjusted. Students are expected to communicate proactively (while the assignment is being worked on) with their tutor regarding issues in contribution, rather than after the assignment is complete.
- **Assignment III (bonus assignment):** Assignment III is an entirely *bonus assignment* that can be used to make up lost marks in the other assignments. Assignment III is designed for students who are looking to extend themselves and are interested in going deeper on the course concepts, working on real-world software. Students who do not wish to do this assignment aren't obliged to, and can simply complete Assignments I and II to achieve satisfactory completion of the course. Assignment III can either be completed in pairs or

individually. Assignment III is due Week 10, Sunday 5pm.

## Final Exam

- There will be a centrally timetabled **in person (face-to-face) final exam** which will be in your UNSW exam timetable. Please note that **all students are required to take the exam in person**, even if they have enrolled in online classes.
- **Hurdle:** In order to pass the course, it is required for the student to achieve a **minimum of 40% (20 out of 50) marks in the final examination**.

## Submission

- All work for the assignments and labs will only be considered for assessment if pushed/uploaded to the platform specified in the assessment specification. This includes that only work pushed/uploaded to the correct platform will be considered for determining contribution levels; and the owner of the account pushing/uploading the work will be deemed the sole author of the work.

## Equitable Learning Plans

- Students who have an Equitable Learning Plan should email their plan to [cs2511@cse.unsw.edu.au](mailto:cs2511@cse.unsw.edu.au) at the beginning of the term. Appropriate allowances to assessments for individual students will be made based on their ELP.

## Assessments

- **Assignment I**, Object-Oriented Design, Due Week 5 Friday, 5pm, 15%
- **Assignment II**, Refactoring & Design Patterns, Due Week 9 Friday, 5pm, 20%
- **Assignment III (not mandatory)**, Event-Driven & Asynchronous Design, Due Week 10 Sunday, 5pm, (Bonus marks up to 8%)
- **Coursework**, All topics, All Weeks, 15%
- **Final Exam**, All Topics, Exam Period, 50%

## Coursework (15%)

- Each week there will be a series of core activities, consisting of a series of Design & Programming lab exercises. In addition to these core activities, students will have the opportunity to undertake choice activities and provide further engagement via the course forum. There are **seven** labs, each worth **ten** marks. We will cap total coursework marks at 60 (which will translate to 15%), leaving one lab as a buffer. If you attend all seven labs, we will add all seven lab marks and cap the total coursework marks to 60. The specific marking criteria for each lab will be outlined in the respective specifications.

## Assignments (35%)

- The marking criteria for the assignments will be outlined in the respective specifications.



## Exam (50%)

- The Final Exam will be held **in person in the CSE Labs, and invigilated** . All the students are required to take the exam in person, even if they have enrolled in online classes.
- **Hurdle** : In order to pass the course, it is required for the student to achieve a minimum of 40% (20 out of 50) marks in the final examination.
- Students are eligible for a Supplementary Exam if and only if: Students cannot attend the final exam due to illness or misadventure. Students must formally apply for a special consideration, and it must be approved by the respective authority.

## Grading Basis

Standard

## Requirements to pass course

**Hurdle:** In order to pass the course, it is required for the student to achieve a minimum of 40% (20 out of 50) marks in the final examination.

# Course Schedule

Teaching Week/Module	Activity Type	Content
Week 0 : 5 February - 11 February		
Week 1 : 12 February - 18 February	Lecture	<ul style="list-style-type: none"><li>• Course Introduction</li><li>• Introduction to Java</li><li>• The Object-Oriented Paradigm</li></ul>
Week 2 : 19 February - 25 February	Lecture	<ul style="list-style-type: none"><li>• Encapsulation</li><li>• Inheritance &amp; Polymorphism</li><li>• Abstract Classes &amp; Interfaces</li><li>• Domain Modelling</li><li>• Design by Contract</li></ul>
Week 3 : 26 February - 3 March	Lecture	<ul style="list-style-type: none"><li>• Exceptions</li><li>• Generics &amp; Collections</li><li>• JUnit Testing</li><li>• Design Principles</li></ul>
Week 4 : 4 March - 10 March	Lecture	<ul style="list-style-type: none"><li>• Refactoring</li><li>• The Functional Paradigm</li><li>• Introduction to Design Patterns</li><li>• Strategy Pattern</li><li>• State Pattern</li></ul>
Week 5 : 11 March - 17 March	Lecture	<ul style="list-style-type: none"><li>• Observer Pattern</li><li>• Refactoring</li><li>• Composite Pattern</li><li>• Test Design</li></ul>
Week 6 : 18 March - 24 March	Lecture	Flexibility Week
Week 7 : 25 March - 31 March	Lecture	<ul style="list-style-type: none"><li>• Creational Patterns</li><li>• Generic Programming</li><li>• Decorator Pattern</li></ul>
Week 8 : 1 April - 7 April	Lecture	<ul style="list-style-type: none"><li>• Event Driven Design</li><li>• Iterator Pattern</li><li>• Template Pattern</li><li>• Adapter Pattern</li><li>• Builder Pattern</li></ul>
Week 9 : 8 April - 14 April		
Week 10 : 15 April - 21 April	Lecture	<ul style="list-style-type: none"><li>• Introduction to Microservice Architecture</li><li>• Course Revision and Wrap-Up</li></ul>

## Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

## General Schedule Information

Please note the following lecture schedule is *subject to change* .

The tutorial and laboratory exercises will be based on lecture content presented in previous weeks.

## Course Resources

### Prescribed Resources

There is no single text book that covers all of the material in this course at the right level of detail. The lectures should provide sufficient detail to introduce topics, and you will then study them in further depth in the labs and assignments.

There are also many online resources available, and we will provide links to the most useful ones. Some are listed below. If you find others, please post links in the Comments section on the Course Outline page.

Some suggestions for books that cover at least some of the topics in this course

- *Head First Design Patterns* , by Elisabeth Freeman and Kathy Sierra, The State University of New Jersey
- *Refactoring: Improving the design of existing code* , by Martin Fowler

## Course Evaluation and Development

This course is evaluated each session using the myExperience system.

However, during the term students are encouraged to provide feedback both during lectures, during tutorials, and generally to course staff via email.

Students are always welcome to provide feedback at any point in the term on their experience by emailing the course account or completing the feedback form linked in the sidebar.

In response to the previous term's feedback, we introduced a sample exam in the Week 10 lab and developed a framework to promote greater engagement during tutorials and laboratories.

# Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Ashesh Mahidadia					No	No
Administrator	Sai Nair					No	Yes
	Alvin Cherk					No	No
	Amanda Lu					No	No
	Carl Buchanan					No	No

## Other Useful Information

### Academic Information

#### I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's [Special Consideration page](#).

#### II. Administrative matters and links

All students are expected to read and be familiar with UNSW guidelines and policies. In particular, students should be familiar with the following:

- [Attendance](#)
- [UNSW Email Address](#)
- [Special Consideration](#)
- [Exams](#)
- [Approved Calculators](#)
- [Academic Honesty and Plagiarism](#)
- [Equitable Learning Services](#)

#### III. Equity and diversity

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

#### **IV. Professional Outcomes and Program Design**

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: <https://www.unsw.edu.au/engineering/student-life/student-resources/program-design>.

*Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.*

#### **Academic Honesty and Plagiarism**

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. *Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.*

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: [student.unsw.edu.au/plagiarism](https://student.unsw.edu.au/plagiarism). The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

[www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf](http://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)

## Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;
- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

## Faculty-specific Information

[Engineering Student Support Services](#) – The Nucleus - enrolment, progression checks, clash requests, course issues or program-related queries

[Engineering Industrial Training](#) – Industrial training questions

[UNSW Study Abroad](#) – study abroad student enquiries (for inbound students)

[UNSW Exchange](#) – student exchange enquiries (for inbound students)

[UNSW Future Students](#) – potential student enquiries e.g. admissions, fees, programs, credit transfer

## Phone

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

## School Contact Information

**CSE Help! - on the Ground Floor of K17**

- For assistance with coursework assessments.

**The Nucleus Student Hub -** <https://nucleus.unsw.edu.au/en/contact-us>

- Course enrolment queries.

**Grievance Officer -** [grievance-officer@cse.unsw.edu.au](mailto:grievance-officer@cse.unsw.edu.au)

- If the course convenor gives an inadequate response to a query or when the course convenor does not respond to a query about assessment.

**Student Reps -** [stureps@cse.unsw.edu.au](mailto:stureps@cse.unsw.edu.au)

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)