Resources / Course Outline

# Course Outline

## 🎉🎉 Welcome to COMP2511 🎉🎉

COMP2511 is a course that builds on the foundation of first-year CSE courses, and where we begin to answer the question "What does good software look like?". We're going to have a great term - even if we're all not physically together. COMP2511 has a teaching team of 25 passionate teaching staff - nearly all of whom have been in your shoes within the last few years. Everyone here will primarily interact with two tutors throughout this term. We're all excited to meet you.

Through the term you'll all be working not only with your tutor(s), but also with our lecturer and course admins.

Ashesh Mahidadia | Braedon Wooding | Nick Patrikeos

This page offers an outline of this course. Take the time to read it, as it covers everything that we expect you from this term, and everything you should expect from us!

*- COMP2511 Teaching Team*

## Contents

## 1. Course Details

| | |
|---|---|
| **Course Code** | COMP2511 |
| **Course Title** | Object-Oriented Design & Programming |
| **Convenor** | Dr Ashesh Mahidadia (https://webcms3.cse.unsw.edu.au/users/z8970925) |
| **Admin Team** | Course Account Email <cs2511@cse.unsw.edu.au><br>**Unless you specifically require to contact a member of the admin team,**<br>**please use the above email for any queries related to the course**<br><br>Braedon Wooding (https://webcms3.cse.unsw.edu.au/users/z5204996)<br>Nick Patrikeos (https://webcms3.cse.unsw.edu.au/users/z5169779) |
| **Classes** | Timetable for all classes (http://timetable.unsw.edu.au/2021/COMP2511.html) |
| **Units of Credit** | 6 |
| **Course Website** | COMP2511 Class Webpage (https://webcms3.cse.unsw.edu.au/COMP2511/21T3/) |
| **Handbook Entry** | http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP2511.html) |

# 2. Course Summary

COMP2511 covers the theory and practice of Object-Oriented Design and Programming with an emphasis on teaching students how to apply software design principles and design patterns to the building of flexible, reusable and maintainable systems. The course also introduces user-interface design and programming and includes the implementation of a large project done in groups using Agile software practices and enables students to have hands-on experience in applying the principles taught, implementing and incorporating the use of good design principles and design patterns in software programs.

# 2.1 Assumed Knowledge

Students are expected to have taken COMP1531 (Software Engineering Fundamentals) and COMP2521 (Data Structures and Algorithms).

It is also assumed that students taking this course are in their second year of study or further, and are comfortable with programming languages and thinking computationally.

More information about specific topics that in both of these courses that are relevant to COMP2511 can be found on this page (https://webcms3.cse.unsw.edu.au/COMP2511/21T3/resources/66520) .

# 2.2 Student Learning Outcomes

The following are the intended learning outcomes of this course:

- *Object-Oriented Design*
  - Understand the principles of Object-Oriented Design
  - Be able to follow a systematic Object-Oriented Design process
  - Be able to interpret and use tools for Object-Oriented Design
  - Learn how to apply design principles and design patterns effectively to design flexible, maintainable and reusable systems

- *Object-Oriented Programming and Java*
    - Understand Object-Oriented Design programming languages
    - Be able to write medium-scale object-oriented programs in Java
    - Apply systematic methods such as test-driven development
    - Understand the importance of documentation and use tools to document software
    - Be able to use an IDE (Interactive Development Environment) for software development
- *Software Processes*
    - Understand the importance of team organization and communication
    - Be able to work within a small team in the context of a software development project
    - Be able to plan and execute a software project according a systematic software process

This course contributes to the development of the following graduate capabilities:

This course contributes to the following UNSW graduate attributes.

- *An in-depth engagement with the relevant disciplinary knowledge in its interdisciplinary context.* The development of systems suitable for different user groups encourages consideration of the users of technology.
- *The capacity for analytical and critical thinking and for creative problem solving.* The creation of software applications requires creative approaches to design and implementation and applying best practices.
- *The ability to engage in independent and reflective learning.* Software design and implementation must be done independently and students benefit by reflecting on their experiences.
- *The skills required for collaborative and multidisciplinary work.* Through a small team project, students gain experience in working within a collaborative environment.
- *The skills of effective communication.* Through a small team project, students develop interpersonal communication skills and presentation and writing skills.

# 3. Teaching Strategies and Rationale

There are some key changes to the structure and content of the course this term. These are strategic changes necessary to improve the student learning experience and outcomes and to adapt to the COVID 19 situation.

## 3.1 Lectures (4 hours each week)

Lectures will be used to present key concepts and practical examples of design techniques taught in this course. Lectures will be delivered via Microsoft Teams Webinar , during weeks 1 to 5, and 7-10.

## 3.2 Tutorials

Every week there is a one hour tutorial, which is scheduled before the lab. Tutorials serve the purpose of:

1. Clarifying ideas from lectures and work through exercises based on the lecture material; and
2. Providing practical demonstrations in the Java programming language.

You should make sure that you use them effectively by examining in advance the material to be covered in each week's tutorial, by asking questions, by offering suggestions and by generally participating.

Tutorials will often involve collaborative and break-out work where you will work with a group of other students to solve or answer problems. This will often be done in your major project group.

Tutorials contribute to your class mark, see the section *Class Mark (10%)* below.

The tutorial questions will be available the week before each tutorial, and the solutions will be available at the end of the corresponding week.

Tutorials/Labs will be run via **MS Teams** . MS Teams is chosen in preference to Blackboard Collaborate due to the relatively higher degree of reliability it provides, the ability to remote control if necessary, and that the features of of Blackboard Collaborate that are useful in other courses (e.g. COMP1511) are not needed in the structure of how COMP2511 is operated.

**All tutorials start in week 1.**

## 3.3 Laboratory Classes & Weekly Labs

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

Each week, there will be a few exercises to work on. These exercises will be released in the week preceding the lab class. The exercises will need to be submitted and will be assessed by your tutor. Lab exercises will constitute a part of the Class Mark, see the section *Class Mark (10%)* below.

All completed lab exercises must be pushed to GitLab and submitted prior to being marked by your tutor or lab assistant.

Labs are due Monday, 1pm of the week following the lab (e.g. the Week 1 Lab is due Week 2 Monday 1pm).

You must then demonstrate your work to your Lab Assistant during the following week's lab. The code that you submit by Monday is what will be assessed. You must not make any changes to the completed solution after Monday. Your Lab Assistant will check the time-stamp prior to marking the lab.

Lab marking will often take up time in tutorials, and your Lab Assistant may not always be able to mark everyone's lab by the end of the 2 hours; any remaining labs to be marked will carry over into the following week. If you would like feedback earlier than this to help with an assessment, you may make a post on the Course Forum and ask for feedback on the lab.

You **cannot** obtain marks by e-mailing lab work to tutors. Your tutor will provide feedback during a lab session on your approach to the problem and on the style of your solution.

Periodically, the lab sessions will also be used to schedule demonstrations of your ongoing group project. As the implementation of the group project will be based on an **agile software development methodology** , you will required to demonstrate progress of your project in Milestones. No excuses will be accepted (unless unavoidable circumstances) for any member of the group not being present for the milestone demo.

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 group project 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 Major Project and the Final Exam.

Use of git and GitLab will be required in this course, as using them for the medium of collaboration and source control is a key learning outcome that will be applied by students in several following courses through their degree.

**All labs start in week 1.**

## 3.4 Help Sessions

Help sessions are unprepared drop-in "clinics" where students and groups can go to seek help about course related matters, whether that be the project, tutorials, or labs. Current tutors or lab assistants will supervise each help session.

Help sessions will be run via Microsoft Teams and using UNSW Hopper.

## 3.5 Assignment

There will be one **assignment** to be completed **individually** which will be released around the end of Week 1 and will be due on Friday of Week 4. This assignment will constitute 15% of the overall course mark.

## 3.6 Project

There will be one main **project** ( **group of 4 students** ) which will run through the rest of the teaching period and contributes to 35% of the overall course mark. You are required to **form your group (of 4)** for the project **by the end of Week 2** .

All group members must be **from the same tutorial / lab** . The specification and other details of the group project will be released at the end of Week 4.

The project consists of three deliverable Milestones. Milestone 1 (5%, Due Week 8) is a formative assessment designed to give groups an opportunity to receive feedback from their tutor on their initial work. Milestone 2 (45%, Due Week 8) and Milestone 3 (50%, Due Week 10) are summative submissions. During the week of the submissions of Milestones 2 and 3, each group will demonstrate their assessment to their tutor during their project check-in time slot.

All groups must satisfactorily complete each Milestone in order to be awarded a mark for the project. Students who fail to contribute to their group will have their individual project mark reduced.

### 3.6 Final Exam

There will be a centrally timetabled final exam which will in your UNSW exam timetable. There is a hurdle requirement on the final exam. **If you do not score at least 40% on the exam, you will not pass the course.** If your overall course score exceeds 50%, despite scoring very poorly (< 40%) on the exam, the hurdle will be enforced via a grade of UF. If your overall course score is less than 50%, then your grade will be FL.

### 3.7 Tools Used in Assessments

All work for the assignment, project, 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.

## 4. Student Conduct

The **Student Code of Conduct** ( Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf) ) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students inform themselves of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

**Plagiarism** is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if you are using these facilities. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and 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.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

# 5. Assessment

| Item | Due | Weighting |
|---|---|---|
| Assignment | Week 4 Friday, 5pm | 15% |
| Project | | 35% |
| -- Project: Milestones 1 + 2 | Week 8 Monday, 1pm | -- 50% of the Project |
| -- Project: Milestone 3 | Week 10 Monday, 1pm | -- 50% of the Project |
| Class Mark (Tutorials + Labs) | All Weeks | 10% |
| Final Exam | Exam Period | 40% |

## 5.1 Class Mark (10%)

Your class mark is made up of marks associated with tutorials, and marks associated with labs. There is a total of 25 marks that can be gained between 9 tutorials and 8 labs throughout the course (i.e. 5 bonus marks), although it will be capped at 20 overall and comprise 10% of your overall course mark.

- You will receive marks out of 2 for each lab. A rough guideline for how marks are awarded for your **lab** is below:
  - 2 / 2 (Completion of all required tasks)
  - 1.5 / 2 (Satisfactory completion of most tasks)
  - 1 / 2 (Did an OK job overall, satisfactory with some errors or items missing)
  - 0.5 / 2 (Completed some things but did so quite poorly)

- You will receive a mark out of one for each tutorial. A number of factors contribute towards you receiving a high mark in this area, including, but not limited to:
    - Attending the full tutorial
    - Asking good and logical questions throughout the tutorial
    - Taking initiative to answer questions and be engaged

You will be notified by the course authority in Week 5 as to how your tutorial attendance and participation mark is progressing.

This marking structure is designed so that students can make choices in how they want to learn. Some students may want to spend time attacking lab exercises, while others want to spend time engaging more in tutorials. With 5 extra marks, there is plenty of room to gain marks.

### 5.2 Assignment

The marking criteria for the assignment will be outlined in the specification which will be released at the end of Week 1.

### 5.3 Project

The marking criteria for the project will be outlined in the specification which will be released at the end of Week 4.

### 5.4 Final Mark

Your final mark for this course will be computed using the above assessments as follows:

```
Course_Work_Mark = lab_mark + assignment_mark + project_mark  (out of 60);
Exam_Mark = Final Exam Mark  (out of 40);
Exam_OK = Exam_Mark >= 16;
Final_Mark = Exam_Mark + Course_Work_Mark;
Final_Grade = UF, if !Exam_OK;
```

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.

## 6. Course Schedule

Please note the following schedule is subject to change.

| Week | Topic | Assessments |
|---|---|---|
| 1 | • Course Introduction<br>• Introduction to Object-Oriented Programming<br>• Introduction to Java<br>• Classes & Objects | |
| 2 | • Inheritance & Polymorphism<br>• Encapsulation<br>• Domain Modelling<br>• Exceptions in Java | Lab 01 Due - Monday 1pm |

| | | |
|---|---|---|
| 3 | • Testing in Java<br>• Java Generics & Collections<br>• Anonymous Classes & Functions<br>• Design Principles | Lab 02 Due - Monday 1pm |
| 4 | • Introduction to Design Patterns<br>• Strategy Pattern<br>• State Pattern<br>• Observer Pattern<br>• Design by Contract | Lab 03 Due - Monday 1pm<br>**Assignment Due - Friday 5pm** |
| 5 | • Agile Software Development & The Project<br>• Testing Theory & Test Design<br>• Composite Pattern<br>• Creational Patterns (Factory Methods) | Lab 04 Due - Friday 5pm |
| 6 | Flexibility Week | |
| 7 | • Singleton Pattern<br>• Introduction to Concurrency<br>• Refactoring<br>• Decorator Pattern | Lab 05 Due - Monday 1pm |
| 8 | • Generic Programming<br>• Iterator Pattern<br>• Template Pattern<br>• Abstract Factory Pattern | **Milestones 1 + 2 Due - Monday 1pm**<br>Lab 07 Due - Tuesday 10am |
| 9 | • Visitor Pattern<br>• Adapter Pattern<br>• Builder Pattern | Lab 08 Due - Monday 1pm |
| 10 | • OOP vs Other Programming Paradigms | **Milestone 3 Due - Monday 1pm**<br>Lab 10 Due - Sunday 5pm |

# 7. Resources for Students

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 group project.

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

# 8. 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.

We have taken on feedback from the 21T2 myExperience, and made changes to the course where possible, including but not limited to the following:

- Decreasing the focus on UI/UX, Frontend Development and Requirements Engineering previously part of the project; and re-adjusting the focus of the course to be more backend-centric;
- Moving to MS Teams for lectures;
- Providing retrospective recordings of some tutorials and lab exercises;
- Providing more support to students via Help Sessions and the course forum; and
- Spending time in discussing Java language features to help students develop good programming practices.

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.

Resource created 2 months ago (Sunday 05 September 2021, 02:31:16 PM), last modified 2 months ago (Wednesday 22 September 2021, 10:31:39 PM).

Comments

🔖 🔍 (/COMP2511/21T3/forums/search?forum_choice=resource/66164)

💬 (/COMP2511/21T3/forums/resource/66164)

There are no comments yet.