



UNSW Course Outline

COMP1531 Software Engineering Fundamentals - 2024

Published on the 15 Feb 2024

General Course Information

Course Code : COMP1531

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

This course teaches students about software engineering principles via exposure to the important practice of building correct products in effectively functioning teams.

You will be exposed to agile software practices, team collaboration and effective communication through implementing a group project based on agile software methodologies that require you to analyse, design, build and deploy a web-based application. This course is typically taken soon after completing COMP1511 but could be delayed and taken later. It provides essential background for the teamwork and project management required in many later courses.

Course Aims

This course aims to provide the students with a strong foundation in the fundamental principles and practices of software engineering that will prepare them for the advanced software engineering workshops. As such, a broad range of key software engineering topics will be taught and reinforced through a group project, that will enable students to apply the theoretical concepts acquired to solve a practical software engineering problem. An agile software delivery style has been chosen for the implementation of the group project, to make students familiar with modern agile development methodologies.

Course Learning Outcomes

Course Learning Outcomes
CL01 : Demonstrate effective use of software development tools to build full-stack end-user applications
CL02 : Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems
CL03 : Understand key characteristics of a functioning team in terms of understanding professional expectations, maintaining healthy relationships, and managing conflict.
CL04 : Demonstrate an ability to analyse complex software systems in terms of their data model, state model, and more.
CL05 : Understand the software engineering life cycle in the context of modern and iterative software development practices in order to elicit requirements, design systems thoughtfully, and implement software correctly.
CL06 : Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties

Course Learning Outcomes	Assessment Item
CL01 : Demonstrate effective use of software development tools to build full-stack end-user applications	<ul style="list-style-type: none"> • Labs • Major Group Project • Individual Project
CL02 : Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems	<ul style="list-style-type: none"> • Labs • Major Group Project • Individual Project
CL03 : Understand key characteristics of a functioning team in terms of understanding professional expectations, maintaining healthy relationships, and managing conflict.	<ul style="list-style-type: none"> • Major Group Project
CL04 : Demonstrate an ability to analyse complex software systems in terms of their data model, state model, and more.	<ul style="list-style-type: none"> • Individual Project • Major Group Project
CL05 : Understand the software engineering life cycle in the context of modern and iterative software development practices in order to elicit requirements, design systems thoughtfully, and implement software correctly.	<ul style="list-style-type: none"> • Individual Project • Major Group Project
CL06 : Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties	<ul style="list-style-type: none"> • Labs • Individual Project • Major Group Project

Learning and Teaching Technologies

CGI

Learning and Teaching in this course

This course uses the standard set of practice-focused teaching strategies employed by most CSE foundational courses:

1. Lectures
2. Tutorials
3. Mega-tutorials
4. Help Sessions
5. Major Group Project

This course aims to provide the students with a strong foundation in the fundamental principles and practices of software engineering that will prepare them for the advanced software engineering workshops. As such, a broad range of key software engineering topics will be taught and reinforced through a group project, that will enable students to apply the theoretical concepts acquired to solve a practical software engineering problem. An agile software delivery style has been chosen for the implementation of the group project, to make students familiar with modern agile development methodologies.

Additional Course Information

Course Schedule

Week 1

????Course Overview

???? Git - Solo Usage

???? Git - Team Usage

???? Javascript

Week 2

???? Package Management

???? Mutli-file & Importing

☒ Dynamic Verification

???? Working as a team

Week 3

???? Data Interchange

???? Continuous Integration

☒ Static Verification

☒ Linting

Week 4

???? Advanced Functions

???? HTTP Servers

Week 5

???? Persistence

???? Iteration 2

☒ Exceptions

???? Designing for Maintainability

Week 7

☒ CodeCoverage

???? Conceptual Modelling

???? Deployment

Week 8

???? Iteration 3

???? Introduction

???? Use Cases, User Stories

???? Validation

Week 9

???? Auth

???? Software Complexity

???? Git - Undo

Week 10

???? (Bonus) Web Front-end

???? (Bonus) Building an MVP

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Labs Assessment Format: Individual	10%	Due Date: TBA
Major Group Project Assessment Format: Group	60%	Due Date: TBA
Individual Project Assessment Format: Individual	30%	Due Date: TBA

Assessment Details

Labs

Assessment Overview

There are 5 assessable labs to submit throughout the course. Each lab consists of 2-3 questions. Each question is worth 1 mark. There are 12 marks attainable across the labs, but you only need to score 10 out of 12 to achieve the full 10% for the lab marking component. This means you can

skip a couple of lab questions throughout the term. Choose wisely!

Each question is completely automarked. After the automarking is complete you will be able to view your marks and the lab solutions.

Lab01 will be marked for you, but those marks won't count toward your final grade. Think of it like a practice lab.

Course Learning Outcomes

- CLO1 : Demonstrate effective use of software development tools to build full-stack end-user applications
- CLO2 : Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems
- CLO6 : Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties

Major Group Project

Assessment Overview

The course staff will form groups of 4-5 students in your allocated tutorials at the beginning of the course.

The project will be implemented using an agile software delivery mode. As such, your team will be required to build and deliver the project in milestones. Each milestone will deliver a part of the requirements of the project and will encompass all the SDLC activities, namely analysis, design, coding and testing. Changes to project requirements are a natural and unavoidable part of any software project life-cycle. Hence, students will need to bear in mind that project requirements may be subject to change and enhancements to functionalities may be made at the end of each milestone. You will need to carefully design the solution for your current milestone, such that the solution is extensible to accommodate these changes.

After certain milestones, your group will present your work in the next lab project check-in that occurs. This is outlined in the major project specification. To receive a mark for that milestone, each team member must be present for the demonstration during the relevant lab time. If working online, this also includes participating with working audio and with their camera on.

Tutors will continually monitor the GitLab repositories to see the team's progress and individual members' contributions to the group project.

The marking criteria for your major project will be specified in the major project specification. In

general, a component of the major project mark is based on automarking and another on manual tutor marking. The project mark is individual-based.

It should be noted that whilst the majority of your mark will come from what is submitted into the deliverables, part of your mark will also be derived from both your individual and your group's participation in tutorials (which includes attending, actively participating, and having a webcam on for online classes).

Feedback for each milestone will be provided in writing and orally in lab project check-ins.

Note that half of the mark for the group project is based on the work by the group as a whole, and half of the mark is based on individual contributions to the group.

Course Learning Outcomes

- CL01 : Demonstrate effective use of software development tools to build full-stack end-user applications
- CL02 : Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems
- CL03 : Understand key characteristics of a functioning team in terms of understanding professional expectations, maintaining healthy relationships, and managing conflict.
- CL04 : Demonstrate an ability to analyse complex software systems in terms of their data model, state model, and more.
- CL05 : Understand the software engineering life cycle in the context of modern and iterative software development practices in order to elicit requirements, design systems thoughtfully, and implement software correctly.
- CL06 : Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties

Individual Project

Assessment Overview

This project will be individual-based. Students will complete a mini project using the knowledge and skills developed in the major project and labs. The mini project will be marked against specific assessment criteria in a marking guide and manual feedback will be provided. The marking is done with automarker and tutor manual marking.

Course Learning Outcomes

- CL01 : Demonstrate effective use of software development tools to build full-stack end-user applications
- CL02 : Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems
- CL04 : Demonstrate an ability to analyse complex software systems in terms of their data

model, state model, and more.

- CLO5 : Understand the software engineering life cycle in the context of modern and iterative software development practices in order to elicit requirements, design systems thoughtfully, and implement software correctly.
- CLO6 : Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties

General Assessment Information

Grading Basis

Standard

Course Schedule

Attendance Requirements

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

Course Resources

Course Evaluation and Development

This course is evaluated each session using the MyExperience system. The following items are being adjusted for 24T1:

1. We have removed the exam or individual assessment to reduce student workload toward the end of term significantly. This has increased the weighting of the major project towards 90% and we've done that will increase the degree to which you are marked individually in the major project.
2. We have re-written all tutorials to improve the experience massively.
3. We have produced mega-tutorials to significantly improve the resourcing we provide to students.
4. Further enhancements to major project spec, automarking, and other quality of life changes.

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Hayden Smith					Yes	No
	COURSE EMAIL					No	Yes

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. 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

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

- 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

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