UNSW College

DPST1093 / CPTG1393 Software Engineering Fundamentals

Term 2 2025

COURSE OUTLINE

DIPLOMA

For detailed information about copyright, ownership and delivery of this course, see page 2.

This course is part of the following programs:

Program Name	CRICOS Course Code
Diploma in Computer Science*	102393G
Diploma of Computer Science^	113046B

* Delivered by UNSW College on behalf of UNSW Sydney - CRICOS Provider Code 00098G; UNSW Sydney TEQSA Provider ID: PRV12055 (Australian University)

^ Delivered by UNSW College under its own CRICOS Provider Code 01020K and TEQSA Provider ID: PRV13020 (Institute of Higher Education)

Copyright Notice

This Course Outline has been made available to you by UNSW Global Pty Limited 2025.

The material in the Course Outline may be subject to copyright under the Copyright Act 1968 (the Act).

Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

TABLE OF CONTENTS

SECTION 1

This Course Outline is designed to guide you through the course successfully. It includes information about the course, your teaching team, how to prepare for classes, details of your assessment tasks, and where to access support and assistance when needed.

Staff

Position	Name	Email
Course Convenor	Dr. Harshana Randeni	H.Randeni@unswcollege.edu.au

Emailing Staff

When contacting staff, you must use your official UNSW College email address. In all emails, please include your:

- ZID
- First and Surname
- Course Name and Course Code

Your teachers are your first point of contact for academic advice or support with your coursework. Their email contact details are easily accessible on your online learning platform. If you have questions about your work, don't hesitate to reach out—they are available to assist you and will respond promptly.

What is expected of you

ATTENDANCE

To achieve academic success, it is recommended that you aim to achieve 100% attendance. You are expected to attend at least 80% of classes. Educational research consistently demonstrates that this level of attendance is associated with a high likelihood of achieving a passing grade.

CHECKING YOUR STUDENT EMAIL

It is important to check your student email account regularly, as all official communications, updates, and announcements related to your course will be sent there. Staying up to date with your email will ensure you do not miss any critical information or deadlines.

ONLINE LEARNING

Access to the course's online learning platform is an essential part of your studies, as all course content will be available online. You are expected to regularly log in to the platform to access lectures, assignments, resources, and updates.

YOUR CONDUCT

You are required to:

- act honestly and uphold academic integrity in all your studies
- treat all students, staff, and affiliates with courtesy and respect
- attend scheduled learning activities and engage actively in your studies
- submit your own work for all assessment tasks and avoid misconduct such as plagiarism or cheating
- use College resources, facilities, and equipment responsibly
- keep your student ID and passwords secure and confidential
- follow all policies, procedures, and lawful directions from staff
- refrain from behaviour that disrupts the teaching and learning environment or negatively impacts others
- maintain the College's reputation through responsible and respectful conduct.

What you can expect from your teachers

TEACHING SUPPORT AND RESPONSIBILITIES

The teaching team is dedicated to supporting your learning and success in this subject. You can expect the teaching team to:

- be well-prepared for each teaching session
- clearly explain course requirements and content
- provide assistance at prearranged times to address your questions or concerns
- deliver constructive and timely feedback on assessments.

PROFESSIONAL CONDUCT

The teaching team is also expected to:

- promptly report any concerns regarding academic or non-academic misconduct
- respond to student queries and concerns about the subject or relevant policies and procedures
- treat all students and staff fairly, respectfully, and in accordance with all policies and procedures.

Changes to this course as a result of student feedback

Student feedback plays a key role in improving this course and we encourage you to participate in any opportunity that allows you to provide feedback, including the My Experience surveys at the end of your teaching session. Changes may be implemented progressively to enhance your learning experience (and those of future students), such as adjustments to teaching materials, assessment tasks, or learning activities. Your input helps ensure the course remains relevant, engaging, and supportive of your academic success. Please note that all surveys are conducted anonymously and, as such, your name or student ID cannot be identified.

Policy and how it affects you

There are multiple policies that apply to students, and it is important to ensure you are referring to the correct policies for the program you are enrolled in. The relevant policies for your program, depending on your enrolment date and program type, are highlighted on the UNSW College website at: https://www.unswcollege.edu.au/about/policies/policy-documents#policies. Please review this page carefully to ensure you are accessing the appropriate policy documents, including:

- Academic Integrity Policy
- Assessment Policy
- Attendance Monitoring Policy
- Enrolment Policy
- IT Security Policy
- Sexual Misconduct Policy
- Student Health, Safety and Wellbeing Policy
- Student Grievances and Complaints Policy
- Student Misconduct Policy

Academic Integrity

Academic integrity requires you to act with honesty, trust, fairness, respect, responsibility, and courage. It is your responsibility to uphold these values by ensuring your work is your own and by appropriately acknowledging the contributions and ideas of others.

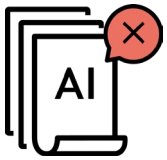
Please adhere to the UNSW College Academic Integrity Policy and Procedure (https://www.unswcollege.edu.au/about/policies/policy-documents#policies) when completing and submitting your assessments.

You are responsible for reading, understanding, and following the instructions about:

- Detecting breaches of academic integrity
- Procedural Fairness
- Investigating a breach of academic integrity
- Outcome of an investigation
- Levels of breach and penalties
- Communications and notifications about academic misconduct
- Recording breaches of academic misconduct
- Appeals

Artificial Intelligence

UNSW College has created its own Artificial Intelligence Guidelines called the 3Cs Model. You are required to adhere to these guidelines when you complete your assessments. These guidelines show you when it is appropriate to use artificial intelligence for your assignment and when it is not permitted.

Category 1: AI tools cannot be used	Category 2: AI tools can be used in assistive role	Category 3: AI tools can be used for integration
		
For this category, you need to show basic skills like remembering, understanding, and using your knowledge, which are essential for future studies and work. For example, you might need to remember and use a formula to solve a math question during a final exam or explain why a business solution is important to your colleagues. In this category, you cannot use AI tools at all. This includes in-person exams, class tests, oral exams, some labs and practicals, and discussion-based assessments. Your teacher will further explain why this category is important.	For this assessment, you may use AI tools to help you develop specific skills. For example, you might use AI to look at data, find patterns in text, and get new ideas about a topic or question. You may use AI tools to help with certain tasks in the assessment. For example, you might use AI to draft and organise content, prepare for exams, test code, translate content, give feedback, and proofread. Your teacher will explain why this category is used to make sure everyone has a fair experience. Some parts of the assessment will not allow AI.	For this assessment, you may use AI tools at different stages of your assessment. In this category, you should show your skills in using AI tools to solve problems, make decisions, and create solutions. This assessment may include a part where you must show that you can use AI in an ethical and responsible way. For example, you might use AI to generate ideas, compare content, produce summaries, analyse content, reframe content, research and find answers, and write content that other students can review and give you feedback on. Your teacher will support and guide you to ensure everyone has a fair experience.

For each assessment in your course guideline, you will see one of the category icons (Category 1, 2 or 3). You can read about the category for your assessment under the heading **Supporting Information**.

If you choose to use artificial intelligence for an assessment that prohibits the use of AI tools, you will be penalised for academic misconduct.

SECTION 2

Course Information

Course code	CPTG1393 / DPST1093
Course name	Software Engineering Fundamentals
Units of credit	6
Course level	Level 1 (Diploma)
Assumed knowledge	CPTG1391 / DPST1091

Course Summary

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 requires you to analyse, design, build and deploy a web-based application. This course provides essential background for the teamwork and project management required in many later courses.

Course Aims

This course aims to provide 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

On successful completion of this course with a passing grade, you will be able to:

1. Demonstrate effective use of applying software development to build backend end-user applications
2. Demonstrate effective use of static testing, dynamic testing, and user testing to validate and verify software systems
3. Understand key characteristics of a functioning team in terms of understanding professional expectations, maintaining healthy relationships, and managing conflict
4. Demonstrate an ability to analyse complex software systems in terms of their data model, state model, and more
5. 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
6. Demonstrate an understanding of how to use version control, continuous integration, and deployment tooling to sustainably integrate code from multiple parties

Program Learning Outcome Alignment

Course Learning Outcome (CLO)	Program Learning Outcome (PLO)	Related Assessment
CLO1: Software Development	PLO1, PLO3, PLO7, PLO11	Major Group Project, Labs
CLO2: Testing Based Development	PLO3, PLO7	Major Group Project, Labs
CLO3: Collaboration	PLO2, PLO5	Major Group Project
CLO4: Conceptual Modelling	PLO1, PLO11	Major Group Project
CLO5: Software Development Life Cycle (SDLC)	PLO3	Major Group Project
CLO6: Software Project Management	PLO5	Major Group Project, Labs

Strategies and Approaches to Learning and Teaching

ATTENDANCE AND ENGAGEMENT REQUIREMENTS

This course will consist of 8 hours of classes each week – 4 hours of lectures (2hr, 1hr, 1hr), 4 hours of combined tutorial-labs (2hr, 2hr). This should be supplemented by 8 hours of self-directed learning, which may include interacting with course content, online activities, research, and work on assessment tasks.

Face-to-face hours	Self-directed hours	Consultation hours
8	8	2

Methodology

A methodology is a roadmap for learning. The roadmap allows you to understand how the learning activities and assessments assist and support learning. The methodologies for this course include:

1. Problem-based learning
2. Design thinking
3. Student-centered learning
4. Teacher-led facilitation
5. Collaborative Group Exercises

METHODS

Lectures: These will focus on introducing concepts and software project management practices. They will occur 3 times a week as a 2 hour, 1 hour and 1 hour session. Attendance is requested but lectures will be recorded to allow for self-paced learning.

Concept focused Tut-labs (2hours): These tut-labs will be held during Wednesday and they will focus on the material covered in the lectures. These tut-labs will also introduce project information to prepare students to collaborate in their group projects. Attendance is required and will affect your project mark.

Practice focused Tut-labs (2hours): These tut-labs will be held during Friday and they will focus on the lab exercises and project progress checks. Attendance is required and will affect your project mark.

LEARNING AND TEACHING ACTIVITIES

Remote check-ins (30mins): These will occur remotely early each week in weeks 4,6,7(twice),8,10,11 and 12. They will allow your project manager to assess your progress and they will give you an opportunity to ask questions and solve blockers.

Help sessions (1hour): These will be remote drop-in sessions that students can join to ask questions about any of the course content

Project Exhibition (1hour): This is an opportunity for students to be acknowledged for all their hard work throughout the term.

Course Schedule and Structure

Week	Lecture/Workshop	Tutorial/Practical	Assessment	Related CLO
1	Course Introduction, JavaScript intro, Git intro	Welcome, C to JavaScript conversion, Git intro		1,5,6
2	Packages, importing files, dynamic verification	Arrays in JavaScript, Code Review, Teamwork, JavaScript +Git	Team formation and Iteration 0 Released and due	1,2,3
3	Data interchange, Continuous Integration, Static verification	Package Management, Testing Procedures	Iteration 1 released	2,4,5,6
4	Linting, Advanced Functions, HTTP Servers	Agile approach, Typing and Typescript, Linting	Iteration 1 due, Peer Review 01 due	2
5	Persistence, Authorization and Authentication, Software Development Lifecycle - Requirements	Intro to APIs, HTTP servers – express server, http tests, Swagger API definitions, First class functions	Iteration 2 Released Iteration 1 demo	1,2,3,5,6
6	SDLC – Use Cases and User Stories, SDLC – Validation, Conceptual modelling, Code coverage	Conceptual modelling using JSON / YAML, Code Refactoring, Using Server Routes		3,4,6
7				
8	SDLC – Maintainability, SDLC – Design Complexity Exception handling, Deployment	Good Software, Code Coverage, System Modelling	Peer Review 02 due Iteration 02 due	1,3,5,6
9	Iteration 3 Focus	Functional vs non-Functional requirements, User Stories and Use Cases, Creating Server Routes	Iteration 2 demo Iteration 3 released	1,2,3,5,6
10	Iteration 3 Focus	Complexity Analysis		2,4
11	Full-Stack – Front end development, Full-Stack – Building a Minimal Viable Product	Project completion focus		3,5,6
12		Course Review Deployment	Iteration 03 due, Iteration 03 demo, Project Exhibition	1,3,5,6

	Peer Review 03 due

SECTION 3

Assessment details

This section outlines the assessment tasks for this course in detail. Use this information as a reference to guide your approach to completing each task.

Assessment Schedule

Task	Weighting	Due date	Course learning outcomes assessed
1. Major Group Project – Iteration 0	3%	Friday 9am, Week 2	1,2,3,4,5,6
2. Major Group Project – Iteration 1	27%	Friday 9am, Week 4	1,2,3,4,5,6
3. Major Group Project – Iteration 2	30%	Friday 9am, Week 8	1,2,3,4,5,6
4. Major Group Project – Iteration 3	30%	Wednesday 9am, Week 12	1,2,3,4,5,6
5. Weekly Lab tasks	10%	Tuesday 9am, Week 2-12	1,2,6
TOTAL	100%		

All marks will be determined in accordance with the UNSW College Assessment Policy and Procedure. You are strongly encouraged to attempt/submit all assessment tasks within this course.

To pass this course, you must achieve an overall mark of 50% or greater

Hurdle: You must participate and contribute in all project iterations

Assessment Guidelines

You are responsible for reading, understanding, and following the instructions about (all information below can be found in the UNSW College Assessment Procedure on this web page https://www.unswcollege.edu.au/about/policies/policy-documents#policies):

- Examinations
- Assessment tasks other than examinations
- Submission and return of assessment tasks, including rules about late assessment submissions
- Feedback on assessment
- Educational adjustments
- Special considerations including applications for extension and deferred examinations
- Supplementary assessments
- Results

ASSESSMENT WORDCOUNT

- You are required to adhere to the word count specified for each assessment.
- A 10% leeway above or below the word limit is permitted without penalty.

- Exceeding this 10% threshold will result in a penalty deduction.
- Details of word count penalties are provided in the assessment section of this Course Outline.

LATE SUBMISSION/COMPLETION OF ASSESSMENTS

Submitting assessments on time is important to ensure fairness for all students. Penalties will apply for late submissions as follows:

- A 5% penalty will be applied for each day the assessment is late, up to a maximum of five days.
- After five days, you will not be able to submit your assessment unless you have an approved educational adjustment or special consideration.
- Penalties are calculated based on the total marks available for the assessment.
- If you submit an assessment more than five days late without approved educational adjustment or special consideration, you will receive a zero mark for that assessment.

Assessment 1 – Major Group Project – Iteration 0

Assessment Type	Group Project – Code and Practice
Weighting	3%
Course learning outcomes addressed	All
Type of collaboration	Group
Task length	30-60 lines of code, 3 documentation tasks, 5 demonstrations of practice
Due date and time	Week 2 Friday 9am
Total Marks	100 marks
AI Category	 AI tools cannot be used
Submission Details	Code Commits to group repository on GitLab, Activities in group repository on GitLab

TASK OVERVIEW

This assessment is focused on learning the tools and systems you will need to complete your major group project. You must carry out the following tasks:

1. Write some stub code in Javascript for the Stage 1 tasks (preparing for Iteration 1)
2. Complete the group contract which outlines the activities you will agree to participate in as a group
3. As a group, you must create an Issue Board for this Group Project on GitLab
4. As a group, you must create a Wiki for this Group Project on GitLab
5. As a group, you must carry out a Data Model walkthrough for the Stage 1 tasks
6. As a group, you must learn to document your group meetings
7. You must create at least one set of commits and one merge request
8. You must create at least one issue for your tasks
9. You must approve another group member's merge request
10. You must set-up your GitLab to connect with your IDE
11. You must set-up your device to be able to run NodeJS locally

RATIONALE

This assessment aims to prepare you for the task of collaborating with other group members for this course. You will be introduced to a new way of running code that differs from Introduction to Programming. You will also be made aware of the various documentation tasks that accompany a successful software development project.

SUPPORTING INFORMATION

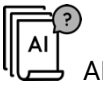
For this stage of the project, since no actual development is being undertaken, the use of AI tools is not allowed. Instead it is recommended that you learn to utilize the various tools and platforms you are introduced to in the first two weeks of the teaching period.

MARKING RUBRIC: MAJOR GROUP PROJECT - ITERATION 0 (3%)

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Function stubs (20 marks)	Everything in a Distinction case but also goes beyond in terms of documentation and organisation. Strictly no functional capability	Meets the highest criteria of the specification in terms of matching expected inputs, outputs and documentation	Meetings the specification in matching expected inputs and outputs and has some documentation	Meets the specification in expected inputs and outputs and has minimal documentation	Does not meet the specification
Group Contract (5 marks)	Everything in Distinction case but also goes beyond in terms of considerations for ongoing Team Collaboration	Meets all the requirements for the group contract laid out in the specification	Meets a substantial number of the requirements of the group contract	Meets at least half of the requirements of the group contract	Does not meet many of the requirements of the group contract
Git Practice (20 marks)	Perfect Git Practice from commit messages, file isolation in commits and documentation	Impressive Git Practice	Good Git Practice	Adequate Git Practice	Sub-standard Git Practice
Merge Practice (20 marks)	Perfect Merge practice from merge requests, approvals, resolutions, conflict handling and documentation	Impressive Merge Practice	Good Merge Practice	Adequate Merge Practice	Sub-standard Merge Practice
Environment Set-up (10 marks)	Perfect environment set-up with a completely isolated IDE that does not rely on external connections	Impressive Environment Set-up	Good Environment Set-up	Adequate Environment Set-up	Sub-standard Environment Set-up

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
GitLab Set-up (10 marks)	Perfect Wiki, Meeting templates, Issue Boards, Milestones and Standards of Practice	Impressive GitLab Set-up	Good GitLab Set-up	Adequate GitLab Set-up	Sub-standard GitLab Set-up
Project Documentation (15 marks)	Perfect Documentation set-up for Code comments, Standards of Practice, Data Model Walkthrough	Impressive Project Documentation	Good Project Documentation	Adequate Project Documentation	Sub-standard Project Documentation

Assessment 2 – Major Group Project – Iteration 1

Assessment Type	Group Project – Code and Practice
Weighting	27%
Course learning outcomes addressed	All
Type of collaboration	Group
Task length	Considerable code development, many documentation tasks and demonstrations of practice
Due date and time	Week 4 Friday 09:00 am
Total Marks	100 marks
AI Category	 AI tools can be used in an assistive role for debugging
Submission Details	Code Commits to group repository on GitLab, Activities in group repository on GitLab

TASK OVERVIEW

This assessment brings you into the focus of how to collaboratively develop a software project. You will be asked to utilize a testing-based development approach to satisfy a software project specification interface. You must out the following tasks:

1. Build on the Standards of Practice you began in Iteration 0
2. As a group, you must develop a shared data model based on the Data Model walkthrough you carried out in Iteration 0
3. As a group, you must develop a function dependency graph – identifying potential common helper functions and any blocking functions
4. As a group, you must use the function dependency graph to separate your tasks into Tiers to allow for better organization
5. As a group, you must distribute all identified tasks from your function dependency group amongst your team members in an equitable manner with appropriate internal deadlines
6. As a group, you must identify potential stakeholders and carry out initial inquiries that relate to the Requirements stage of the Software Development Life Cycle
7. You must develop tests for your intended development functions based on the specification interface
8. You must create appropriate tasks and update those tasks on the Issue Board to track your progress
9. You must attend and participate in check-ins with your project manager twice-weekly – one near the beginning of the week and one in-class meeting during your Friday tut-lab
10. You must complete your assigned coding tasks within the internal deadlines and integrate them into your group's codebase using appropriate git and merge practice

RATIONALE

The objective of this assessment is to develop your skills in how to approach a software project that begins with client requirements somewhat established. You will learn how to break down a large project into manageable tasks and collaborate as a team to complete these tasks. You will learn about the software development life cycle and its various steps and how to use it to identify relevant stakeholders to your software project. You will also learn the practice of testing-based development as this course does not have an auto-testing framework similar to Introduction to Programming as you will be developing your own tests. This approach also matches industry best practices.

SUPPORTING INFORMATION

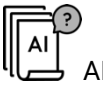
In this stage of the project, you may use AI assistants for debugging your code. You may not use the output of any GenAI system as a submission. In most cases, it is very evident when you use GenAI content for code or tests. It is also difficult to integrate with your team members work. Your check-in and tut-lab attendance and participation will also affect your mark for this assessment.

MARKING RUBRIC: MAJOR GROUP PROJECT – ITERATION 1 (27%)

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Function Automarking (30 marks)	The marks for this criteria are exactly reflective of the number of autotests passed				Any hard coding will result in a 0 being awarded for this section
Testing (20 marks)	Perfectly executed tests for this iteration in terms of coverage, organisation and design	Impressive test design and clarity with perfect coverage	Impressive test design and clarity with less than perfect coverage	Adequate test design and clarity	
Code quality (10 marks)	Perfect production ready code with immaculate styling and use of Javascript-specific concepts	Impressive Code quality	Good code quality	Adequate code quality	Sub-standard code quality
Git Practice (10 marks)	Perfect adherence to conventional commits guidelines, commit file isolations and documentation	Impressive Git Practice	Good Git Practice	Adequate Git Practice	Sub-standard git practice
Merge Practice (10 marks)	Perfect Merge practice from merge requests, approvals, resolutions, conflict handling and documentation	Impressive Merge practice	Good Merge practice	Adequate Merge practice	Sub-standard Merge practice
Software Development Life Cycle (5 marks)	Perfect identification and articulation of stakeholders and preparing for Requirements analysis	Impressive attempt at identifying stakeholders and preparing for requirements analysis	Good attempt at identifying stakeholders and preparing for requirements analysis	Adequate attempt at identifying stakeholders and preparing for requirements analysis	Sub-standard attempt at identifying stakeholders and preparing for requirements analysis

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Project Documentation (15 marks)	Perfect documentation of code, group practices, meetings, issue and wiki management	Impressive attempt at project documentation	Good attempt at project documentation	Adequate attempt at project documentation	Sub-standard attempt at project documentation

Assessment 3 – Major Group Project – Iteration 2

Assessment Type	Group Project – Code and Practice
Weighting	30%
Course learning outcomes addressed	All
Type of collaboration	Group
Task length	Considerable code development, many documentation tasks and demonstrations of practice
Due date and time	Week 8 Friday 09:00am
Total Marks	100 marks
AI Category	 AI tools can be used in an assistive role for debugging
Submission Details	Code Commits to group repository on GitLab, Activities in group repository on GitLab

TASK OVERVIEW

This assessment builds on the standards of practice of the previous two assessments and introduces two key new aspects – Typescript as a development choice over Javascript; and HTTP Servers. You will also learn to structure a project milestone in terms of sprints. This involves setting more stringent internal deadlines, organizing sprint leaders and extending the standards of practice from the previous iteration. You must carry out the following tasks:

1. Build on the Standards of Practice from Iteration 0 and 1
2. As a group, you must understand the new specification format
3. As a group, you must map your new specification HTTP routes to your previous iteration functions
4. As a group, you must develop a shared data model based on a new Data Model walkthrough for this Iteration
5. As a group, you must develop a function dependency graph – identifying potential common helper functions and any blocking functions
6. As a group, you must use the function dependency graph to separate your tasks into Tiers to allow for better organization
7. As a group, you must organize your project tasks into Sprints with internal deadlines within each sprint
8. As a group, you must carry out Software Development Life Cycle tasks related to the Requirements stage in preparation for designing and presenting new features in your next Iteration.
9. You must act as sprint leader for at least one of your sprints for this Iteration or the next.
10. As a sprint leader, you must guide your group for your sprint, including asking questions from your project manager, staying on top of merge request approvals and preparing a sprint hand-off document for the next sprint leader
11. You must develop HTTP tests for your intended development functions based on the specification interface
12. You must create appropriate tasks and update those tasks on the Issue Board to track your progress
13. You must attend and participate in check-ins with your project manager twice-weekly – one near the beginning of the week and one in-class meeting during your Friday tut-lab
14. You must complete your assigned coding tasks within the internal sprint deadlines and integrate them into your group's codebase using appropriate git and merge practice

RATIONALE

This assessment will give you a familiar understanding of HTTP communication and the beginnings of full stack development. You will develop skills related to leading your project team and organizing a more extensive software project. You will also learn the importance of carrying out the various techniques associated with the Requirements Stage of the Software Development Life Cycle.

SUPPORTING INFORMATION

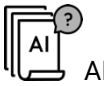
In this stage of the project, you may use AI assistants for debugging your code. You may not use the output of any GenAI system as a submission. In most cases, it is very evident when you use GenAI content for code or tests. It is also difficult to integrate with your team members work. Your check-in and tut-lab attendance and participation will also affect your mark for this assessment.

MARKING RUBRIC: MAJOR GROUP PROJECT – ITERATION 2 (30%)

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Function Automarking (35 marks)	The marks for this criteria are exactly reflective of the number of autotests passed				Any hard coding will result in a 0 being awarded for this section
Sprint Planning (10 marks)	Perfect organisation of all Project tasks into Sprints, all deadlines adhered to, all documentation completed, all preparation completed for each sprint handoff	Impressive Sprint planning	Good Sprint Planning	Adequate Sprint Planning	Sub-Standard Sprint Planning
Code Features (10 marks)	Perfect production ready implementation of Auth, Persistence and Security	Impressive attempts at code features	Good attempts at code features	Adequate attempts at code features	Sub-standard attempts at code features
Software Development Life Cycle (10 marks)	Perfectly carry out Industry level requirements analysis, elicitation, User Story identification, Use Case definition and User Acceptance Criteria creation	Impressive attempt at SDLC Requirements tasks	Good attempt at SDLC Requirements tasks	Adequate attempt at SDLC Requirements tasks	Sub-Standard attempt at SDLC Requirements tasks
HTTP Testing (15 marks)	Perfectly executed HTTP tests for this iteration in terms of coverage, organisation and design	Impressive attempt at writing HTTP Tests	Good attempt at writing HTTP Tests	Adequate attempt at writing HTTP Tests	Sub-Standard attempt at writing HTTP Tests

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Standards of Practice (20 marks)	Maintain Perfect Standards of practice for Git, Merge, Code Quality and Project Documentation established in previous Iterations	Impressive levels of Standards of Practice	Good levels of Standards of Practice	Adequate levels of Standards of Practice	Sub-Standard levels of Standards of Practice

Assessment 4 – Major Group Project – Iteration 3

Assessment Type	Group Project – Code and Practice
Weighting	30%
Course learning outcomes addressed	All
Type of collaboration	Group
Task length	Considerable code development, many documentation tasks and demonstrations of practice, a 3 minute pitch presentation and a 2 minute video
Due date and time	Week 12 Wednesday 09:00am
Total Marks	100 marks
AI Category	 AI tools can be used in an assistive role for debugging  AI tools can be used for the LLM-specific task
Submission Details	Code Commits to group repository on GitLab, Activities in group repository on GitLab

TASK OVERVIEW

This assessment builds on the all the standards of practice established in the previous iterations and introduces the concept of a Finite State machine. You will also complete SDLC Design related tasks to design new functionality for your project. You will carry out the following tasks:

1. Build on the Standards of Practice from Iteration 0,1 and 2
2. As a group, you must understand the new specification format
3. As a group, you must develop a shared data model based on a new Data Model walkthrough for this Iteration
4. As a group, you must develop a function dependency graph – identifying potential common helper functions and any blocking functions
5. As a group, you must use the function dependency graph to separate your tasks into Tiers to allow for better organization
6. As a group, you must organize your project tasks into Sprints with internal deadlines within each sprint
7. As a group, you must carry out Software Development Life Cycle tasks related to the Design stage in preparation for designing and presenting new features in your next Iteration.
8. As a group, understand how to implement a State Diagram into the execution of your code
9. You must act as sprint leader for at least one sprint if you have not previously done so.
10. As a sprint leader, you must guide your group for your sprint, including asking questions from your project manager, staying on top of merge request approvals and preparing a sprint hand-off document for the next sprint leader
11. You must develop HTTP tests for your intended development functions based on the specification interface
12. You must create appropriate tasks and update those tasks on the Issue Board to track your progress
13. You must attend and participate in check-ins with your project manager twice-weekly – one near the beginning of the week and one in-class meeting during your Friday tut-lab
14. You must complete your assigned coding tasks within the internal sprint deadlines and integrate them into your group's codebase using appropriate git and merge practice

RATIONALE

This assessment will bring you an understanding of how State is managed in software projects. You will learn to adapt your codebase to a changing specification as is often the case in industry software projects. You will learn to carry out design related tasks of the Software Development Life Cycle and you will learn how to present these ideas to stakeholders. There will also be bonus tasks which will utilize LLM APIs for advanced students.

SUPPORTING INFORMATION

In this stage of the project, you may use AI assistants for debugging your code. You may not use the output of any GenAI system as a submission. In most cases, it is very evident when you use GenAI content for code or tests. It is also difficult to integrate with your team members work. Your check-in and tut-lab attendance and participation will also affect your mark for this assessment.

In addition, you may use an LLM API that we provide to practice prompt engineering and mini-agent design.

MARKING RUBRIC: MAJOR GROUP PROJECT – ITERATION 3 (30%)

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Function Automarking (45 marks)	The marks for this criteria are exactly reflective of the number of autotests passed				Any hard coding will result in a 0 being awarded for this section
Sprint Planning (10 marks)	Perfect organisation of all Project tasks into Sprints, all deadlines adhered to, all documentation completed, all preparation completed for each sprint handoff	Impressive Sprint planning	Good Sprint Planning	Adequate Sprint Planning	Sub-Standard Sprint Planning
Software Development Life Cycle (10 marks)	Perfectly carry out SDLC Design tasks and present to stakeholders in a manner that would get instant approval in industry	Impressive attempt at SDLC Design tasks	Good attempt at SDLC Design tasks	Adequate attempt at SDLC Design tasks	Sub-Standard attempt at SDLC Design tasks
HTTP Testing (15 marks)	Perfectly executed HTTP tests for this iteration in terms of coverage, organisation and design	Impressive attempt at writing HTTP Tests	Good attempt at writing HTTP Tests	Adequate attempt at writing HTTP Tests	Sub-Standard attempt at writing HTTP Tests
Standards of Practice (20 marks)	Maintain Perfect Standards of practice for Git, Merge, Code Quality and Project Documentation established in previous Iterations	Impressive levels of Standards of Practice	Good levels of Standards of Practice	Adequate levels of Standards of Practice	Sub-Standard levels of Standards of Practice

Assessment 5 – Weekly Labs

Assessment Type	Weekly Labs
Weighting	10%
Course learning outcomes addressed	CLO1: Software Development CLO2: Testing Based Development CLO6: Software Project Management
Type of collaboration	Individual
Task length	1-4 hours work per task, 15 tasks total
Due date and time	Week X Friday 11.59 pm
Total Marks	1 mark each
AI Category	 AI tools cannot be used
Submission Details	Submission through Turnitin link in Moodle.

TASK OVERVIEW

These tasks will assess the theory and practice concepts that are introduced in the lectures. There are 15 tasks in total, each task is worth up to 1 mark. You will receive a cumulative mark capped at 10.

RATIONALE

You should attempt these tasks on your own to gain the skills and capabilities you need to complete and collaborate in the group project.

SUPPORTING INFORMATION

You should avoid using AI in this assessment as it will not aid you in your understanding while learning the principal elements of the programming language and software project management practice. If you use AI from the beginning, then you will be unable to determine if it is hallucinating or not and you will be unable to integrate code into larger projects.

MARKING RUBRIC: WEEKLY LABS (10%)

Assessment Criteria	High distinction (85-100%)	Distinction (75-84%)	Credit (65-74%)	Pass (50-64%)	Fail (0-49%)
Function Automarking (all marks)	The marks for this criteria are exactly reflective of the number of autotests passed				Any hard coding will result in a 0 being awarded for this section

SECTION 4

Essential Resources for Students

Essential resources are the materials and technologies you will require to complete this course. The essential resources for this course include:

1. Microsoft Office 365 suite as provided by the College
2. A laptop (Electronic Device Requirements)

Support for Students

UNSW College offers a range of support services to help you with your studies and student experience:

Resources	
UNSW College Current Students Website	A wide range of resources are available to support your academic and personal success. These include study advice, time management tools, course planning assistance, and academic support services. The Current Student Website also provides quick links to essential information, such as: • Health and Wellbeing resources • Academic support and learning tools • Student news and events • Forms and key administrative services Visit the Current Student Website at https://my.unswcollege.edu.au/ to access everything you need in one place.
Teacher Consultations	Schedule time with your teachers for study advice and subject-related support.
Study Club	Book a place through your online learning platform under Student Consultations to join group study sessions.
Student Advisers	Located at the Student Services Office (Level 1, Building L5), Student Advisers can assist with time management, study skills, course planning, and personal issues affecting your studies. Book appointments online via the Support section of the UNSW College Student Hub.
General Assistance	For help with matters such as accommodation, student activities, or personal concerns, visit the Student Services Office.
Subject Consultations	Times and locations will be provided by your teachers or can be found on the online learning platform.

Take advantage of these services to support your learning and overall success at UNSW College.

References

This course does not use any prescribed texts or materials.

Last amended:	May, 2025

ACKNOWLEDGEMENTS

Contributors: Harshana Randeni