COMP2511
The Art of Software Design
(OO Design & Programming)

Course Introduction

Term 1, 2025

Our Team

Lecturer-in-charge:

- Dr Ashesh Mahidadia <a.mahidadia@unsw.edu.au>

Course Admin Team:

- Alvin Cherk
- Sai Nair
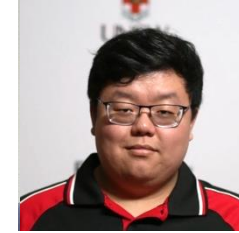- Michael Mospan
- Grace Kan
- Daniel Khuu

Tutors:

- 25+ passionate tutors!

Course Account Email: cs2511@cse.unsw.edu.au
*(Unless you specifically require to contact a member of the admin team, please use the **above email** for any queries related to the course.)*
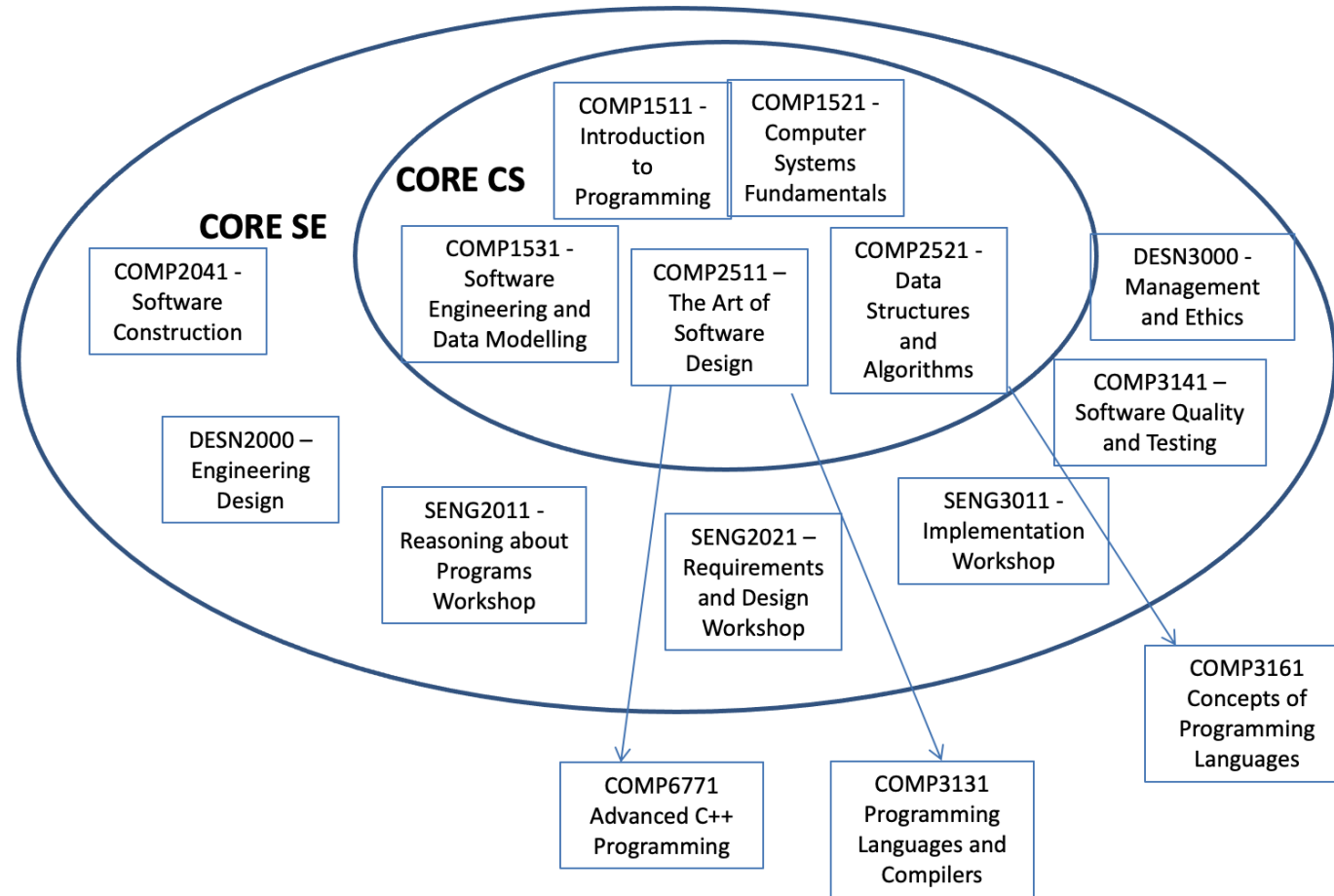
Class Web: http://webcms3.cse.unsw.edu.au/COMP2511/25T1/

Course Context

CORE SE

CORE CS

COMP1511 - Introduction to Programming

COMP1521 - Computer Systems Fundamentals

COMP2041 - Software Construction

COMP1531 - Software Engineering and Data Modelling

COMP2511 – The Art of Software Design

COMP2521 - Data Structures and Algorithms

DESN3000 - Management and Ethics

COMP3141 – Software Quality and Testing

DESN2000 – Engineering Design

SENG2011 - Reasoning about Programs Workshop

SENG2021 – Requirements and Design Workshop

SENG3011 - Implementation Workshop

COMP3161 Concepts of Programming Languages

COMP6771 Advanced C++ Programming

COMP3131 Programming Languages and Compilers

The Story So Far: Course Context

- COMP1511: Solving problems with computers, the wonder and joy of programming

- COMP1521: Getting right down into the silicon

- COMP1531: Solving problems in a team; programming in the large

- COMP2521: Solving problems at scale using data structures and algorithms

- **COMP2511???**

COMP2511

- We can write code, but how do we write good code?

- **Designing elegant and beautiful software.**

- Shades of Grey - things aren't clear cut; writing good software is an art

- From programmers to designers.

COMP 2511 Major Themes

- Develop an appreciation for elegantly written software, and how to create and maintain well-designed systems;

  - Apply principles and patterns to effectively design **flexible**, **maintainable** and **reusable systems**

- Understand different design paradigms and methodologies, their background and application;

  - Object-Oriented Paradigm

  - Functional Paradigm

  - Concurrent Paradigm (introduction)

COMP 2511 Major Themes

- Understand and apply the principles of Object-Oriented Design to solve problems;
  - Be able to follow a **systematic** OO Design process
  - Be able to interpret and use tools for OO Design
- Understand the role of and apply widely used Design Patterns to create extensible designs
  - Behavioural patterns
  - Structural patterns
  - Creational patterns
  - Programming patterns (exceptions, generic programming)
  - Testing patterns

COMP 2511 Major Themes

- Develop skills in both creating medium-scale systems from scratch, and working on existing systems as part of the Software Development Life Cycle;

  - Be able to analyse, refactor and work with code started by someone else

  - Create medium-scale systems using Java

- Work with an enterprise programming language and IDE

  - Java language

  - VSCode IDE

Credit teaching material

❖ No text book, the lecture slides cover the required topics.

❖ However, you are strongly encouraged to read additional material and the reference books.

❖ In the lecture notes, some content and ideas are drawn from:

- *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
- Material from many popular websites.

How do we obtain our educational objectives?

❖ **Lectures:** 4 hour lectures (9 weeks)

❖ **Tutorials:**

    ❖ A 1 hour tutorial session per week, which is scheduled before the lab.

    ❖ Online Tutorials/Labs will be run via **MS Teams** .

    ❖ Tutorials are understanding-driven - interactive examples to illustrate concepts discussed in lectures

    ❖ Solutions and recording to tutorials posted at the end of each week

How do we obtain our educational objectives?

❖ **Labs:**

    ❖ 2 hours each week, straight after tutorial

    ❖ Similar to most CSE core courses

    ❖ Lab retros posted at after due date on Confluence

    ❖ Online Run via MS Teams

The above are subject to change, if required.

Assessments

Coursework (15%)

❖ Your coursework mark is made up of marks associated with the lab exercises.

❖ 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.

❖ The table below offers as a general guide for the criteria that your tutor/lab assistant will use to assess you.

    o A general guide for lab assessments

Assignment I (15%)

❖ The marking criteria for the assignment will be outlined in the specification which will be released Tuesday of Week 2.

❖ Due Friday 5pm Week 5.

❖ Completed **individually**.

Assignment II (20 %)

- ❖ The marking criteria for the project will be outlined in the specification which will be released Thursday Week 5.

- ❖ **Pairs** formed within your tutorial.

- ❖ Groups formed by end of Week 3.

- ❖ Due **Friday 5pm week 9**

- ❖ Measurers in place for difficult partners (Keep your tutor informed)

Assignment III (8 % Bonus)

❖ A more challenging real-world problem that incorporates Deign Principles and Patterns discussed in the course.

❖ For students that wish to extend themselves and score highly in the course

❖ Can be completed in a pair or individually

❖ Assignment spec released **Tuesday week 8**

❖ Due **Sunday 5pm of week 10**

Final Exam (50%)

❖ In 25T1 the COMP2511 exam will be **held in person in the CSE Labs, and invigilated**.

❖ **All** the students are required to take the **final exam in person**, even if they have enrolled in online classes. In 25T1, there will be no online exams.

❖ *Hurdle* : From Term 3 2023, 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.

Assumed Knowledge

❖ Confident programmers

  o Familiar with C and Python programming concepts

❖ Able to work in a team

  o Git

  o Working with others

❖ Understand basic testing principles

❖ Understand basic software engineering design principles (DRY, KISS)

Assumed Knowledge

❖ What we don't assume:

- o Knowledge of Java

- o Understanding of Object-Oriented Programming

❖ **This is not a Java course**

Course philosophy

❖ A step up from first year courses

❖ Challenging but achievable

❖ Develop skills in time management, teamwork as well as critical thinking

❖ Highly rewarding

Support

❖ Supporting you is our job :)

❖ Help Sessions

    o Lots of them with fantastic tutors

    o Feedback on work, help with problems, clarifying ideas

    o You are expected to have done your own research and debugging before arriving

Support

❖ Course Forum (Ed)

    o Ask questions and everyone can see the answers!

    o Make private posts for sharing code

    o Response time

❖ Course Account - cs2511@cse.unsw.edu.au

    o Sensitive/personal information

❖ During the project - your tutor

Support

❖ Go to help sessions for help on concepts

❖ Post on the forum if you need more immediate lab feedback

❖ There are no late extensions on labs unless in extenuating circumstances - email cs2511@cse.unsw.edu.au

Support - UNSW

❖ **Special Consideration** -
https://student.unsw.edu.au/special-consideration

❖ **Equitable Learning Services** -
https://student.unsw.edu.au/els

Mental Health & Wellbeing

❖ UNSW Psychology & Wellness - https://student.unsw.edu.au/mhc

❖ UNSW Student Advisors - https://student.unsw.edu.au/advisors

❖ Reach out to us at cs2511@cse.unsw.edu.au

❖ Check in with each other

❖ Talk to someone

Technology Stack

❖ Java Version – JDK 17

❖ VSCode

❖ Gradle 8.5

❖ Gitlab (+ CI pipelines)

Feedback

❖ We love feedback :)

❖ Changes made to the course this term based on constructive student feedback

❖ We always want to continuously improve

   o 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.

❖ Feedback form

❖ Course account

Respect

❖ Yourselves, each other, course staff

It's time to lift off for 24T3!!!!