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

Course Introduction

Term 1, 2023

Our Team

Ashesh Mahidadia

Lecturer in Charge

Sienna Archer

Course Admin

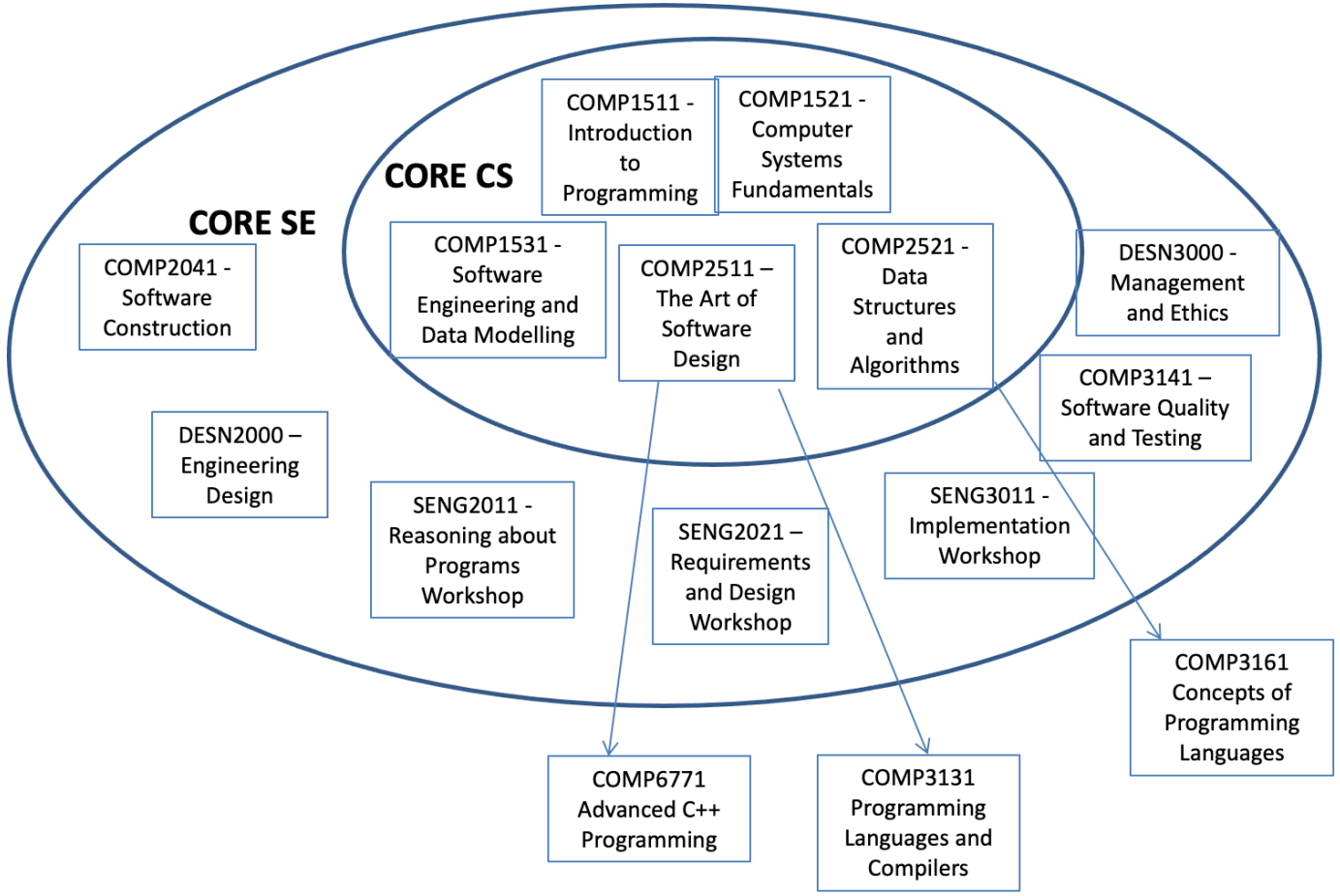
Lecturer-in-charge:
Dr Ashesh Mahidadia <a.mahidadia@unsw.edu.au>

Course Admin:
Sienna Archer <sienna.archer@unsw.edu.au>

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/COMP2511/23T1/

Course Context



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
  - Online Run via MS Teams
  - Lab retros posted at after due date on Confluence

- **Weeks 1 - 5**:
  - Lab marking: Tutors will mark your completed labs
  - Help/assistance with labs/assignment
- **Weeks 7 – 8**:
  - Lab marking
  - Assignment I vivas
- **Week 9:**
  - Lab marking (Last week to get labs marked)
- **Week 10:**
  - Assignment II vivas

Assessments

Class Mark (15%)

- Your class mark is made up of marks associated with the lab exercises. Bonus Forum marks!
- Your final class mark is out of 48/56.
- Your class mark for each week will be out of 8. Meaning you get one labs worth of marks as buffer.
- There are additional choice activities which students may complete if they wish. Highly recommended you go and finish everything as study for the exam.
  They are non-assessable.
- Forum Participation (Only towards Class mark)
  - 5+ questions answered across the term - 1 mark
  - 10+ questions answered across the term - 2 marks
  - 20+ questions answered across the term - 3 marks
  - 40+ questions answered across the term - 4 marks
  - More than 50 questions answered across the term - 5 marks

Assignment I (20%)

- 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**.
- Oral assessment (interview) on your work in your week 7 or 8 lab (If you cant attend either week you will need to apply for special consideration)

Assignment II (30 %)

- 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)
- Oral assessment in week 10 lab

Assignment III (10 % 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 corse
- Can be completed in a pair or individually
- Assignment spec released **Tuesday week 8**
- Due **Sunday 5pm of week 10**

Exam (35%)

- In 23T1 the COMP2511 exam will be held entirely **IN PERSON,** in the cse labs and invigilated.
- If you cannot take the final (**IN PERSON**) exam because you are overseas, you need to apply for a special consideration and take the supplementary (**IN PERSON**) exam. If that is not suitable, you should enroll in a following term when you can take an in-person exam.
- 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
  - Familiar with C and Python programming concepts
- Able to work in a team
  - Git
  - Working with others
- Understand basic testing principles
- Understand basic software engineering design principles (DRY, KISS)

Assumed Knowledge

- What we don't assume:
  - Knowledge of Java
  - 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
  - Lots of them with fantastic tutors
  - Feedback on work, help with problems, clarifying ideas
  - You are expected to have done your own research and debugging before arriving

Support

- Course Forum (Ed)
  - Ask questions and everyone can see the answers!
  - Make private posts for sharing code
  - Response time
- Course Account - cs2511@cse.unsw.edu.au
  - 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 11
- VSCode
- Gradle 5.4.1
- 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
  - Providing simpler alternatives to some of the more complex labs in the course to enable students are more concise lab learning experience and moving from 8 labs to 7;
  - Reducing the workload of Assignment II;
  - Adjusting the Assignment I deadline to enable students to gain more experience and understanding of OO Design concepts before submitting Assignment I;
  - Moving Assignment III from 5% core, 5% bonus to 0% core, 10% bonus;
- Feedback form
- Course account
- Student representatives

Respect

- Yourselves, each other, course staff

Let's have a fantastic T1!!!