

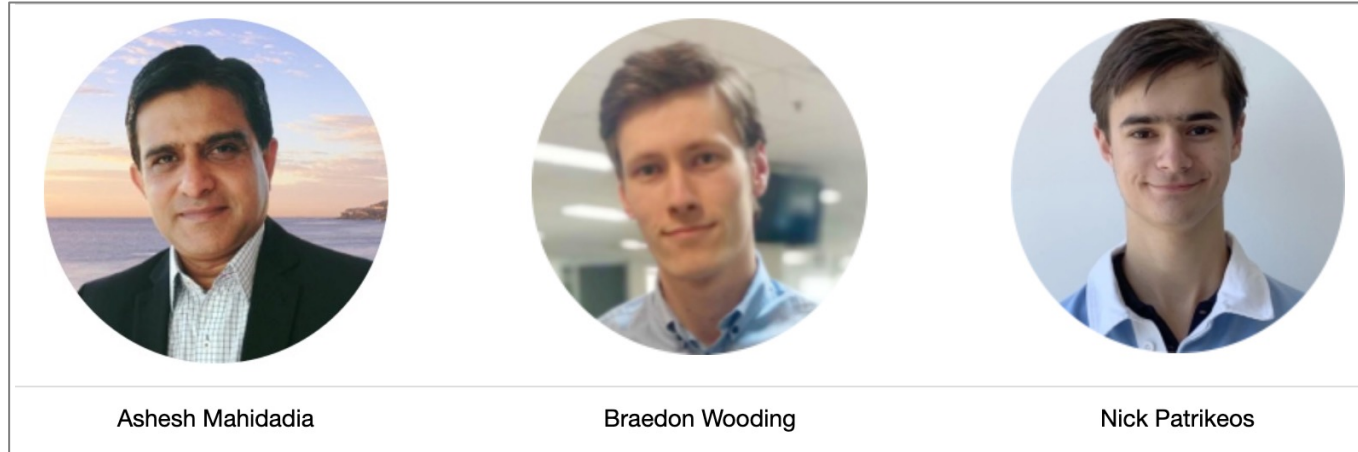
# COMP2511

# Object Oriented Design & Programming

## Course Introduction

Term 3, 2021

# Our Team



## Lecturer-in-charge:

Dr Ashesh Mahidadia <ashesh@cse.unsw.edu.au>

## Course Admin Team:

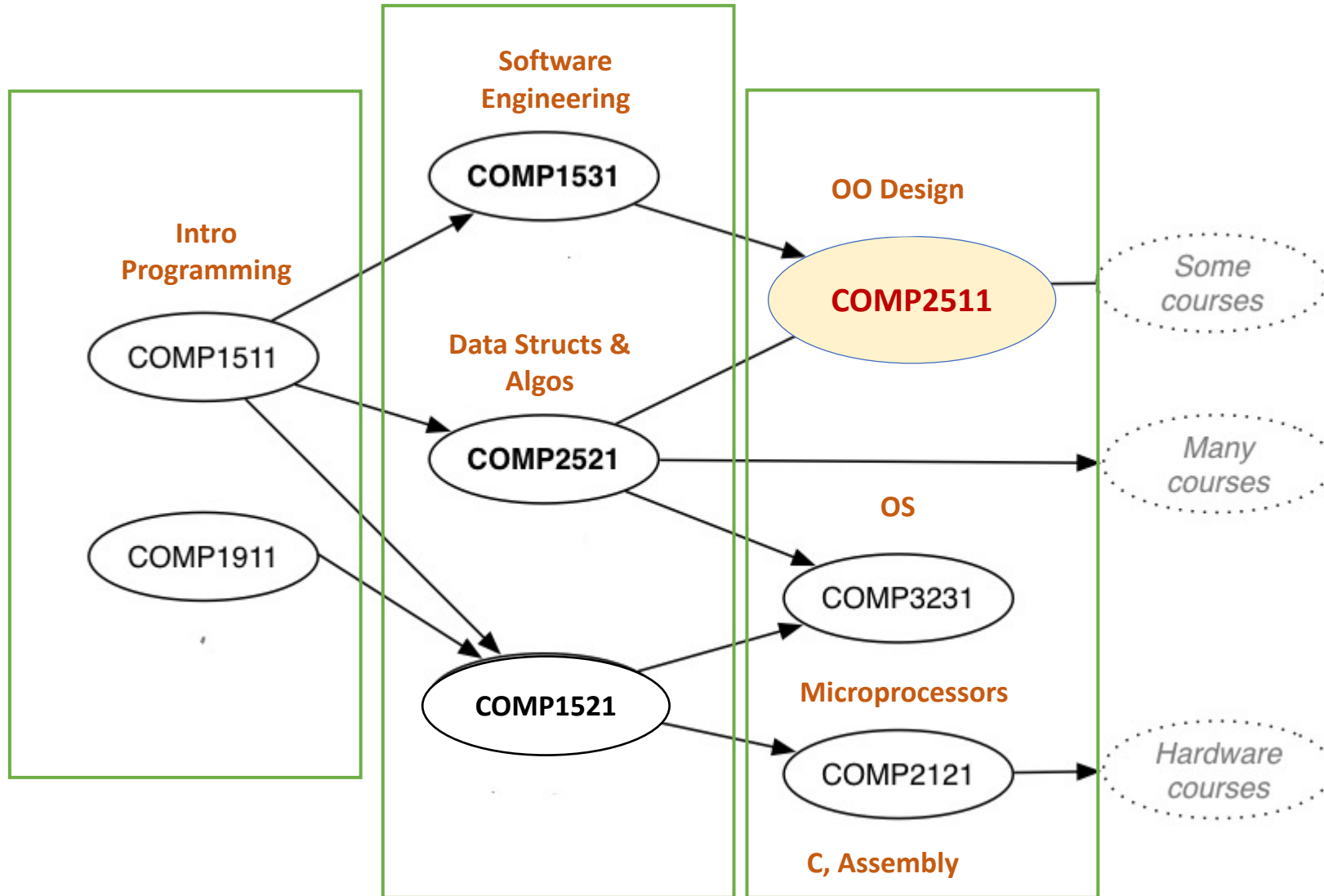
Braedon Wooding <b.wooding@unsw.edu.au>,  
Nick Patrikeos <n.patrikeos@unsw.edu.au>,

**Course Account Email:** [cs2511@cse.unsw.edu.au](mailto: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/COMP2511/21T3/>

# 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.
- From programmers to designers.



# COMP 2511 Major Themes

## ❖ Object Oriented Design Process (SE)

- 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

# COMP 2511 Major Themes

## ❖ Object Oriented Programming in Java

- Be able to write medium-scale object-oriented programs in Java

## ❖ Software Engineering process

- Problem solving – apply SE principles to solve a real-world problem
  - 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

# 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.
- ❖ Tutorials/Labs will be run via **MS Teams** .
- ❖ Tutorials contribute to your class marks.
- ❖ You will receive mark out of one for each tutorial. A number of factors contribute toward you receiving a high grade 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
- ❖ What if I can't make it??
- ❖ 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
- ❖ Run via MS Teams
- ❖ **Weeks 1 - 4:**
  - ❖ Lab marking: Tutors will mark your completed labs
  - ❖ Help/assistance with labs/assignment
- ❖ **Weeks 5 - 10:**
  - ❖ Lab marking
  - ❖ Project check-ins: Group mentoring sessions with your tutor
  - ❖ Weeks 8 + 10: Demonstrations of milestone submissions

# How do we obtain our educational objectives?

## ❖ Labs:

- ❖ Labs contribute to your class marks.
- ❖ 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 marks (completion of all required tasks)
  - 1.5 / 2 marks (satisfactory completion most tasks)
  - 1.0 / 2 marks (did an OK job overall, satisfactory with some errors or items missing)
  - 0.5 / 2 marks (completed some things but did so quite poorly)
  - 0.25 / 2 marks (barely an attempt)

## ❖ Class Marks (tut/lab marks):

- ❖ 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. 4 bonus marks), although it will be capped at 20 overall and comprise 10% of your overall course mark.

# 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**

# Assessments

# Class Mark (10%)

- Tutorial Participation - up to 9 marks available
- Weekly labs:
  - Give the skills needed to complete assignment and project.
  - Enable critical thinking.
  - Each lab is worth 2 marks
- 25 marks available, class mark is capped at 20.
- Labs are (usually) due the following Monday, 1pm.
- Labs released the week prior.
- Lab solutions not be released, some have retrospective videos.

# Assignment (15%)

- Due Friday Week 4
- Completed individually
- Spec released end of Week 1



# Project (35%)

- ❖ Groups of 4 formed within your tutorial
- ❖ Groups formed by **end of Week 2**
- ❖ Spec released end of Week 4
- ❖ Structure
  - ❖ Milestone 1: Formative feedback (anytime)
  - ❖ Milestone 2: Submission of work (Week 8 Monday)
  - ❖ Milestone 3: Submission of work (Week 10 Monday)
- ❖ Measures in place to ensure a difficult group doesn't take a toll on marks

## Exam (40%)

- ❖ Hurdle - must achieve at least 40% in the exam to pass the course
- ❖ Tests course concepts taught throughout the term

# 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](mailto: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](mailto: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](mailto:cs2511@cse.unsw.edu.au)
- ❖ Check in with each other
- ❖ Talk to someone



# Feedback

- ❖ We love feedback :)
- ❖ Changes made to the course this term based on constructive student feedback
- ❖ We always want to continuously improve
- ❖ Feedback form
- ❖ Course account
- ❖ Student representatives

# Respect

❖ Yourself, each other, course staff

Let's have a fantastic T3!!!

