

COMP 2511 (Term 2, 2019)

**Object Oriented Design &
Programming**

Introduction

Our Team

Lecturer-in-charge:

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

Course Admin:

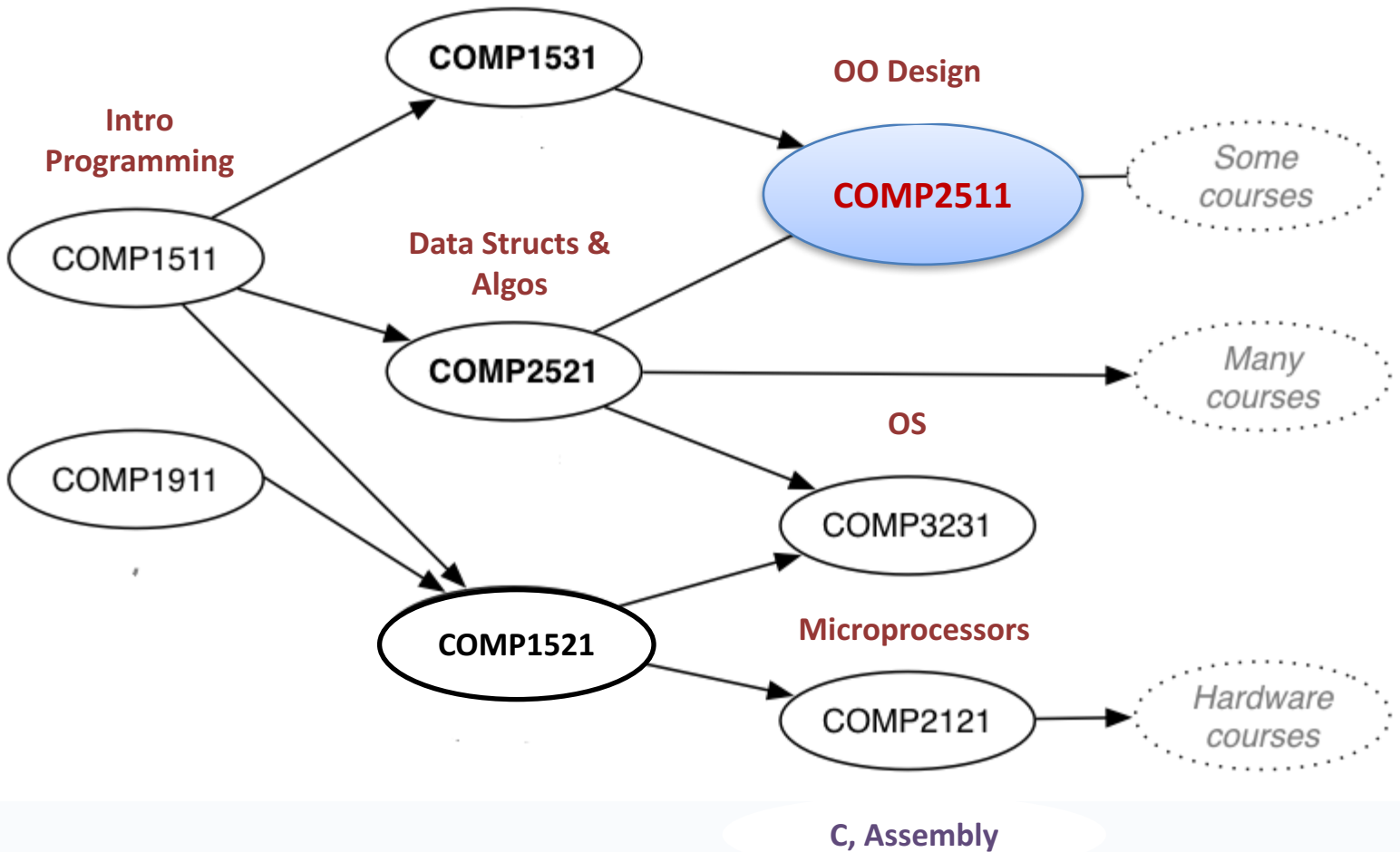
Robert-Clifton Everest
cs2511@cse.unsw.edu.au

Class Web:

<http://webcms3.cse.unsw.edu.au/COMP2511/19T2/>

Course Context

Software
Engineering



COMP 2511 thoughts

- What have you heard about COMP2511?
- Why take COMP2511?

Pre-requisites

- Completed **COMP 1511**
 - competent C programmers who can understand and use abstract data types
- Completed **COMP 1531**
 - understand and know how to use git and Github reasonably
 - be familiar with fundamental object-oriented design concepts
- Some of you may have not completed COMP 1531
 - Week 1 lab to familiarise with git/GitLab
 - Additional bridging material (particularly in weeks 1,2)

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

• User Interfaces

- Effective coding and testing techniques

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
- ❖ Links to useful tutorials will be uploaded as necessary

How do we obtain our educational objectives?

❖ Lectures... (Recorded)

- 4 hour lectures

❖ Tutorials and Labs...

- 3 hours per week (1 hour tut followed by lab), weeks 1 – 10
- Tutorials re-enforce learning from lectures
- Labs comprise small design and practical programming exercises, individual or in pairs

❖ A Group Project...

- Project iteration demos scheduled in some lab sessions

Assessments

- Tutorial & Lab Sessions: 3 hours per week (10%)
 - Tutorials re-enforce learning from lectures and provide practical demonstrations
 - Marks will be awarded for completion of lab exercises
 - Lab sessions encompass both design and practical programming exercises
 - To obtain any lab marks for a Week X lab, you must:
 - *push your completed solution to GitLab and submit it*
 - *demonstrate your work to your tutor in the lab class in the week it is due OR in the first hour of the following week (completed solution must be submitted by Sunday 5 PM of Week X)*
 - **Cannot** obtain marks by emailing solution to tutors

Assessments

- Assignment 1 (10%)
 - Specification will be released at the end of Week-1
- Final Exam (55%)
 - 3 hour final exam, more details through the term

Group Project

- Contributes to **25%** of the final course mark
- Carried out in teams of **2**
- Project specification will be released in Week 04
- Implemented using an *Agile Software Development Model*
 - Working software to be delivered in iterations
 - Marks will be awarded for each iteration demo, which will count towards your overall group project mark
 - Responsibilities to be assigned to each group member during each iteration and all team members **MUST** contribute equally (**Tutors will check GitLab**)
 - Final project demo held in week 10

Course Mark

- Course Work Mark (out of 45) =
Lab (10%) + Assignment 1 (10%) + Group Project (25%)
- Exam Mark = Mark from the 3 hour final exam (out of 55)
ExamOK = (Exam Mark \geq 22.5/55)
- Final Course Mark (out of 100) =
Course Work Mark + Exam Mark
- Final Grade
 - **UF**, If **Not(ExamOK)** , even if final course mark > 50
 - UNSW grade, otherwise

Supplementary Exam

- Students are eligible for a Supplementary Exam if :
 - they cannot attend the final exam due to illness or misadventure
 - successfully **apply** for a **special consideration** (must apply for a special consideration, and get approved)
 - a supplementary exam will **not** be awarded for any other reason.

System

- ❖ Most work done on Linux or Mac
 - Lab work and group project can be done on the CSE machine labs or your own device
 - Technology stack
 - Java 8 SE, Eclipse
 - Use Eclipse
- ❖ Collaboration and Versioning Tool - GitLab

Plagiarism



Just don't do it!