COMP2511
Object Oriented Design & Programming

Course Introduction

20T3

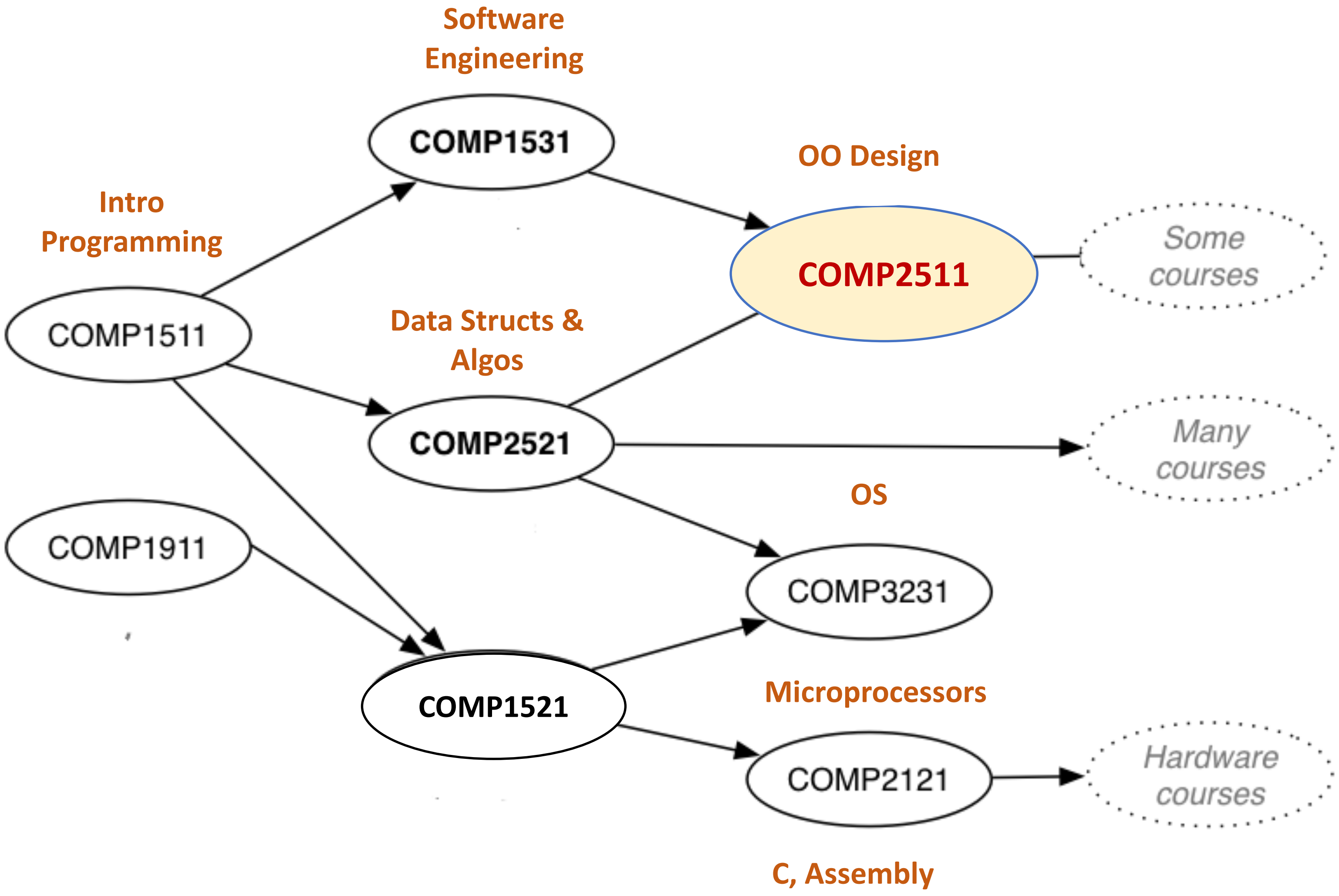
Our Team

Lecturer-in-charge:
Dr Ashesh Mahidadia
ashesh@cse.unsw.edu.au

Course Admin:
Matthew Perry
cs2511@cse.unsw.edu.au

Class Web:
http://webcms3.cse.unsw.edu.au/COMP2511/20T3/

Course Context



Intro Programming

Software Engineering

Data Structs & Algos

OO Design

OS

Microprocessors

C, Assembly

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

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…**

- 4 hour lectures (9 weeks)

❖ **Tutorial (recording)…** 1 hour each week (no tutorial in week-6)

- watch a 1-hour tutorial recording prior to attending your scheduled tutorial
- Your tutor or lab assistant will be available during your tutorial time-slot to answer any questions you may have regarding the tutorial questions after watching the tutorial recording.

❖ **Labs…**

- 2 hours per week, weeks 1-5 and 7 – 10
- Labs comprise small design and practical programming exercises, individual or in pairs

❖ **A Group Project…**

- Project iteration demos scheduled in some lab sessions

Tutorials

Tutorial recording - 1 hour each week

You will be expected to watch a 1-hour tutorial recording prior to attending your scheduled tutorial Q&A session each week. This video will discuss questions from that week's tutorial content.

Tutorials Q&A - 1 hour each week

Your tutor or lab assistant will be available during your tutorial time-slot to answer any questions you may have regarding the tutorial questions after watching the tutorial recording.

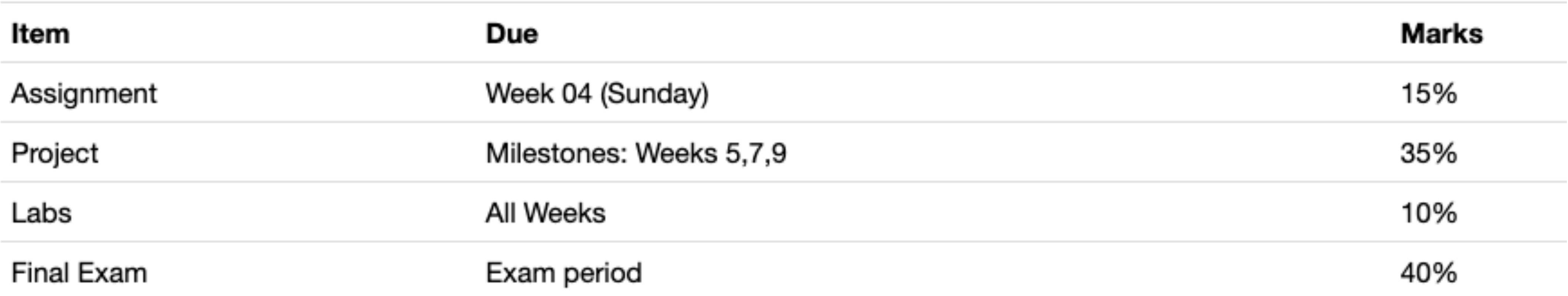
Laboratory classes

- There are six labs (weeks 1, 2, 3, 5, 7 and 9) with specified lab exercises.
- Your tutor or lab assistant will be available during your lab time-slot to answer any questions you may have regarding the lab questions.
- The other labs will be used for lab and project assessments.
- Please note that you must attend all nine lab and tutorial sessions during the term.
- You should **demonstrate** your work to your tutor during the lab or the following week's lab for feedback on your problem solving, approach, the style of your solution and importantly **marking** .

Summary: you must:

- *demonstrate your work to your tutor in the week X lab class* (for feedback and marking)
- OR *demonstrate your work at the start of the lab in week X+1* (for feedback and marking)

Assessments

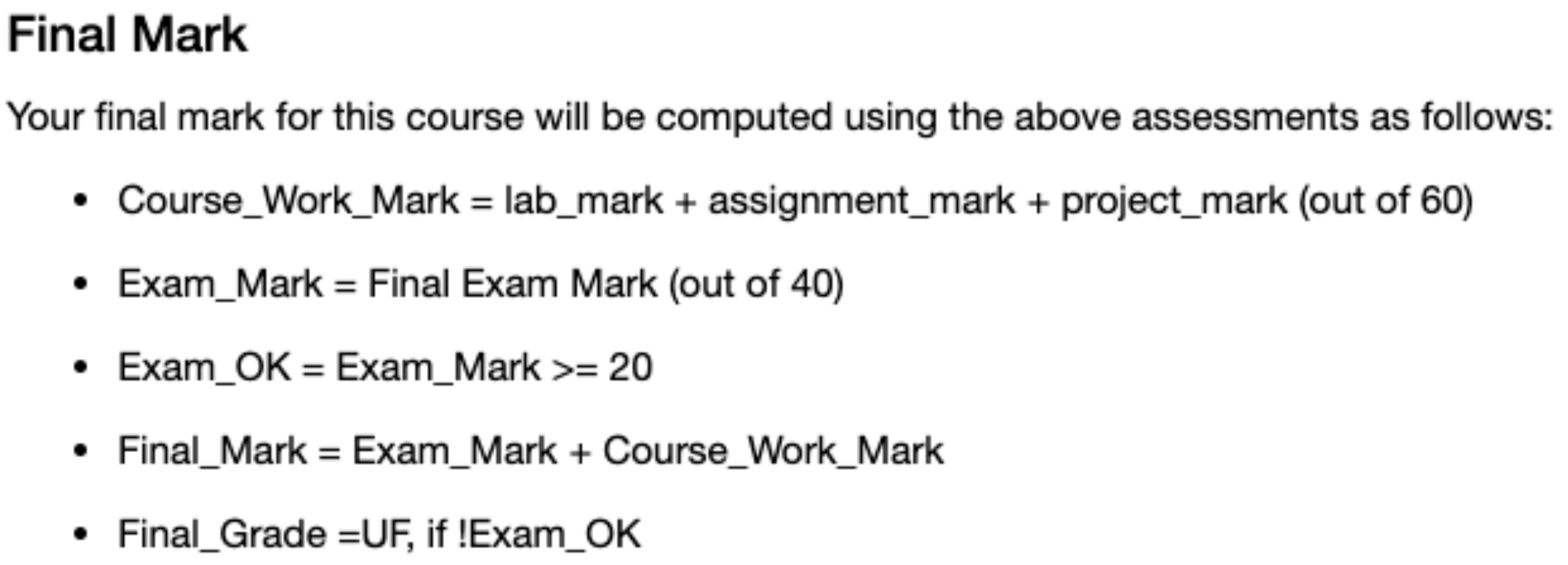


Item	Due	Marks
Assignment	Week 04 (Sunday)	15%
Project	Milestones: Weeks 5,7,9	35%
Labs	All Weeks	10%
Final Exam	Exam period	40%

Group Project

- Contributes to 35% of the final course mark
- Carried out in teams of **two**
- 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



Final Mark

Your final mark for this course will be computed using the above assessments as follows:

- Course_Work_Mark = lab_mark + assignment_mark + project_mark (out of 60)
- Exam_Mark = Final Exam Mark (out of 40)
- Exam_OK = Exam_Mark >= 20
- Final_Mark = Exam_Mark + Course_Work_Mark
- Final_Grade =UF, if !Exam_OK

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**)
  - For more information, read *Essential Advice for CSE Students*, the web link is available in the course outline.

https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students

System

- Most work done on Linux or Mac
  - Lab work and group project can be done on the CSE machines (using vlab) or your own device
  - Technology stack
    - Java 11 SE,
    - Visual Studio Code
- Collaboration and Versioning Tool - GitLab

Plagiarism

Just don’t do it!