Dr Jake Renzella, Dr Yuchao Jiang

UNSW SYDNEY

COMP1531
Software Engineering Fundamentals

Teaching Strategies - WebCMS for Info

- lectures
- Tutorials
- Labs
- Major Project
- Help Sessions

Getting Help

If you need help throughout the course, seek it in this order:

Step 0: Your team
Step 1: EdStem forum (Webcms3 sidebar)
Look for answers before posting
You were invited “z5555555@student.unsw.edu.au”
Step 3: Help Sessions
Step 4: Emailing Tutor / Assistant Tutor
Step 5: Lecturers cs1531@cse.unsw.edu.au

👷 Getting Setup

Any operating system is fine for this course.

- CSE Machines (VLab / SSH): Ready to go!
- Linux: Follow getting started guided
- Mac OSX: Follow getting started guided
- Windows: Follow getting started guided

Full instructions relating to the install can be found in our guide here

😻 Technology In The Course

The key technologies we use in the course are:

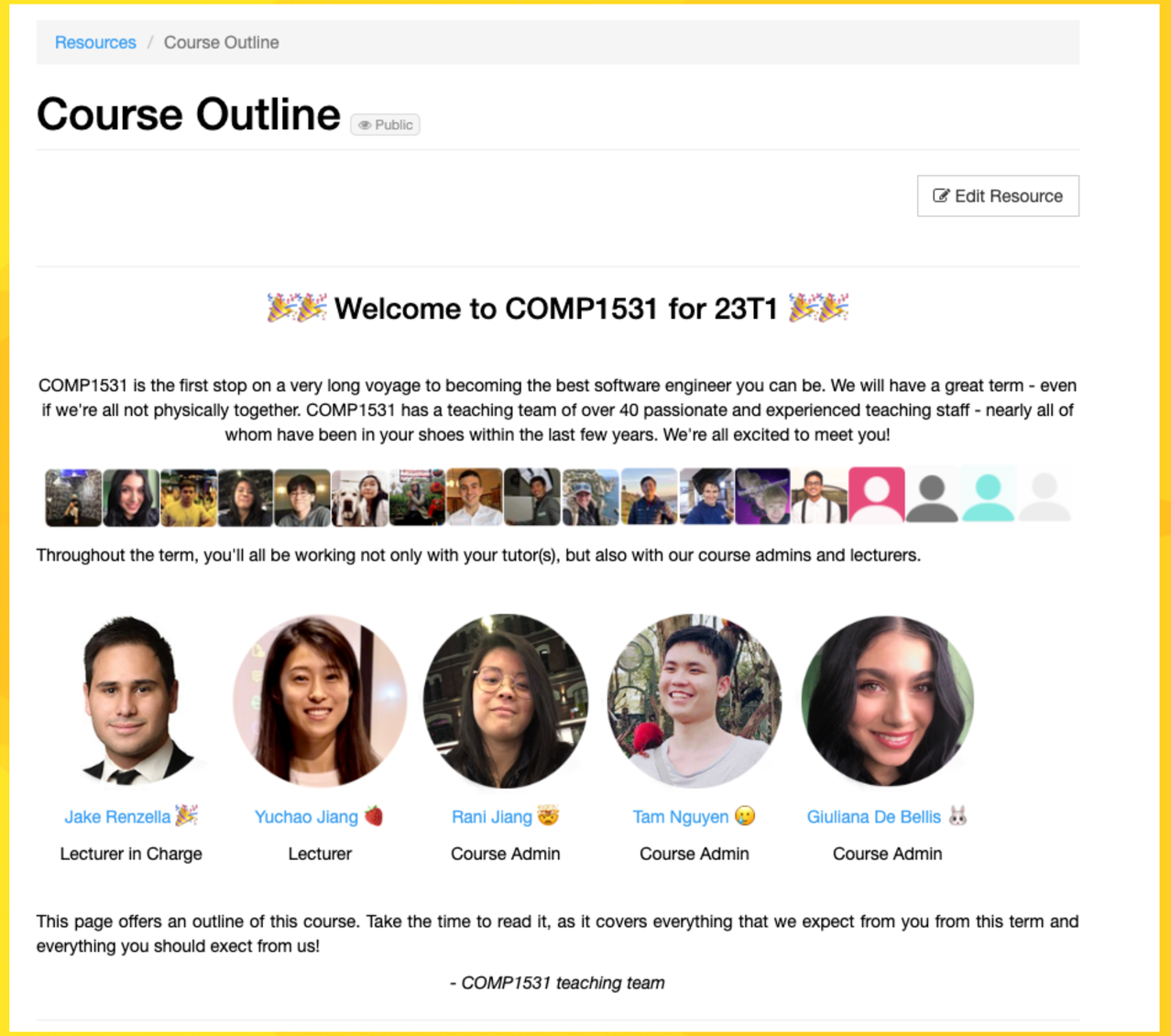
- Webcms3 (our home!)
- EdStem (forum)
- Gitlab (where your work lives)
- Microsoft teams (for project communication and online class)

The key pieces of developer software we use are:

- git (GIT version control)
- node (Javascript - NodeJS)

Week-1 Checklist

- Access WebCMS and read Course Outline
- Access EdStem Forum (and post in the Welcome thread)
- Go through the getting started page on WebCMS
- Access Gitlab



What is Software Engineering?

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (IEEE 1990)

COMP1511: Learning to write code

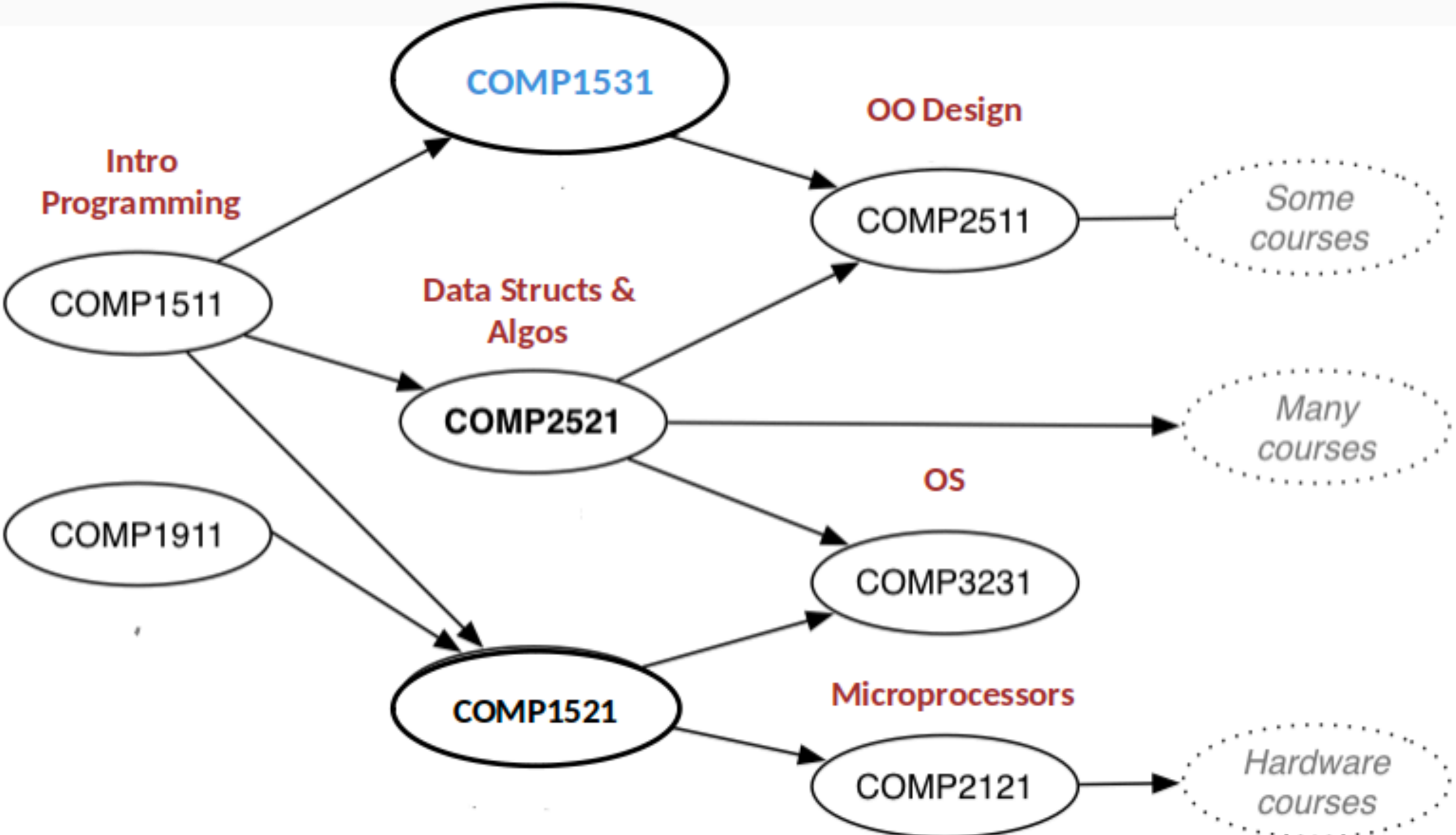
COMP1531: Learning to design, architect, test and building larger software systems

Computer Science: understanding how computers work and function - i.e. making a computer do things

Software Engineering: focus on building the right software for the right people, and making sure it's maintainable over time as it grows and people working on it change.

Do you need to be a Software Engineering student to be a Software Engineer?

- Backend/Frontend/Full-stack Engineers
- DevOps Engineers
- Security Engineers
- Software/QA Testers


COMP1531
OO Design
Intro Programming
COMP2511
Some courses
COMP1511
Data Structs & Algos
COMP2521
Many courses
OS
COMP1911
COMP3231
COMP1521
Microprocessors
COMP2121
Hardware courses

We want all of you to be good software engineers. That means we want you to be able to:

🔎 Extract and understand the problem you're trying to solve.
🛠️ Design and develop code that resists regressions when it changes over time (problem or people).
✅ Ensure that software is as safe as possible by verifying that it's correct.
🌏 Understand that the purpose of software is to make it available for use.
👬 Work together with a group of people.

Good software engineers don't always write the most optimal solution...

- They write good code that works, and is maintainable by others
- They learn to avoid costly mistakes

We're Going To Have Fun!


Requirements


Design


Development


Correctness


Projects


Coding together


Full-stack

These topics are all about helping us iterate through the software development life cycle whilst building quality software and working well with others.

🔎 Requirements

Understanding a problem we're trying to solve.

If we don't do this right, we might build something that works well for the wrong purpose. We learn about requirements engineering, use cases, and validation.

🎨 Design

Good design is about setting up good architecture for our system but also trying to write elegant code. We'll try and model systems and their complexity as well as write graceful and thoughtful code.

🛠️ Development

We need to actually code "the thing", and in this course we're using Javascript. We'll learn about Javascript's language features in multi-file projects.

✅ Correctness

Software needs to be verifiably correct as much as possible. We need confidence that it's doing "the thing" correctly. We'll learn about testing code statically and dynamically, exceptions, code coverage, and linting.

🌏 Projects

Code lives on more than just your computer. Code you use will be written by others, and code you write will need to be given to others. We learn about how to manage multi-package projects, and ensure that we have a continuous and automated way to integrate multiple people's code and deploy that to "users".

👬 Coding Together

Teamwork is both about people and about tools. We'll learn about how to work with version control software between people, as well as how to function as a group of programmers.

Full-Stack

Many software projects are now web-based and have frontends and backends. We'll be working in similar systems using HTTP and data layers to write a functioning server.

Lecture Feedback