Course Outline

Contents

- 1. Course Details
- 2. Course Summary
- 3. Philosophy of teaching web-based front-ends
- 4. Teaching Strategies
- 5. Assessment
- 6. Schedule / Timetable
- 7. Student Conduct
- 8. Resources for Students

1. Course Details

Find information relating COVID-19 and this offering here (https://www.covid-19.unsw.edu.au/safe-return-campus-faqs) and here (https://edtech.eng.unsw.edu.au/c19mess/comms.html) .

Course Code	COMP6080
Course Title	Web Front-end Programming
Convenor	Hayden Smith (https://webcms3.cse.unsw.edu.au/users/z3418003)
Admin	Hayden Smith (https://webcms3.cse.unsw.edu.au/users/z3418003)
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs6080/21T3/ (http://cse.unsw.edu.au/~cs6080/21T3/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6080.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6080.html)

2. Course Summary

COMP6080 aims to develop your confidence in building modern web-based applications to industry standards. This occurs through the introduction of a range of basic concepts surrounding HTML, CSS, Vanilla Javascript, Javascript Declarative Frameworks, UI/UX Principles, Accessibility, Network & Asynchronous Programming, Front-end Testing, and other basic infrastructure.

This course has a heavy emphasis on industry voices, and as such a number of lectures (approximately 33% of the course) will be given by current front-end developers from industry. For 21T3 the bulk of this industry support comes from Canva (https://canva.com/) , with some less major help from Google (https://lmgtfy.com/?q=google) .

2.1. Assumed Knowledge

COMP6080 assumes that you have satisfactorily completed COMP1531 and COMP2521. The main areas of knowledge you'll need from these courses are:

- A high level understanding of how interpreted scripting languages work (e.g. python), in terms of inputs, interpretation, loose typing, etc
- Confident use of GIT as a version management tool
- Basics of HTTP protocol and interacting with web browsers

Postgraduate students will need to familiarise themselves with git, specifically, if not already confident, check out the resources here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60815) . We have also provided a lab0 on the tutorials & exercises page (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60837) that we encourage everyone who hasn't completed COMP1531 to complete prior to the beginning of the course. Basic competency in git is an expectation in COMP6080.

2.2. Student Learning Outcomes

After completing this course, students will be able to:

1. A knowledge of fundamental Javascript semantics sufficient to design, construct, test and debug Javascript programs.
2. An understanding of the context of web-front end programming, including CSS and the DOM, sufficient to successfully construction of programs for the web-front end.
3. A knowledge of modern web front-end coding practices including Javascript frameworks and CSS framework, allowing effective and efficient integration of existing commonly-used components into a web application.
4. An ability to use modern Javascript asynchronous programming techniques for the web-front end to produce robust and maintainable code, and a basis to build on in understanding concurrency in other contexts.
5. A knowledge of basic web front-end security issues sufficient to understand and avoid common security vulnerabilities when constructing web applications.
6. An awareness of user interface design as it applies to the web, including accessibility, allowing adoption of sound practices and providing a basis to build an understanding of interface design generally.

---

3. Philosophy of teaching web-based front-end

3.1. Challenges of teaching front-end

The challenges of teaching front-end in a finite time period are related to the size and scope of knowledge around modern web front-end. Building even the most basic modern web app requires an extreme breadth and depth of abstractions that take typical developers years to become very comfortable with.

Teaching front-end is also challenging due to the rapid advances that are made in the world of web (a good thing), which leads to sprawling and conflicting resources across the internet. This can lead to learning inconsistent or sub-standard practices and programming patterns.

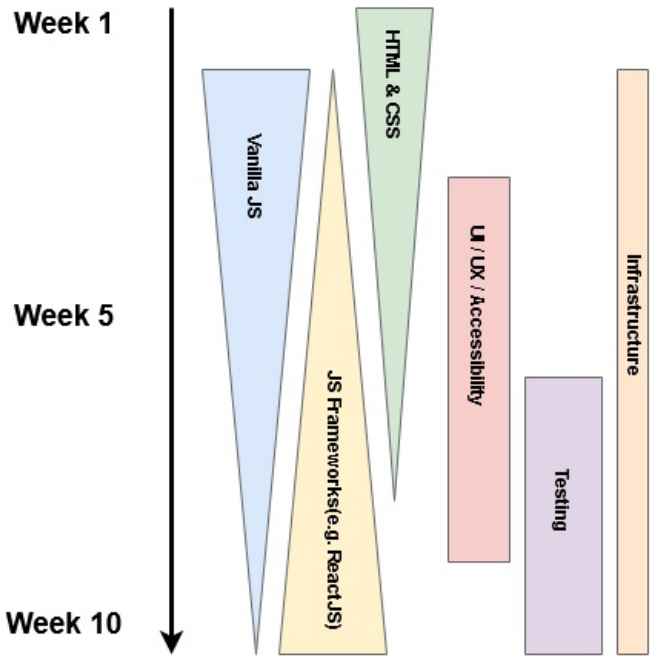
To teach only declarative frameworks (e.g. ReactJS) as an introduction creates the problem of producing capable students who can rapidly produce work, but do so without the fundamental understanding to grow and apply their skills sustainably and with accountability. However, to teach only Vanilla Javascript would neglect the reality that nearly all students in this course will inevitably not be building actual applications with Vanilla Javascript.

3.2. Approach to teaching

This course is about spending a term learning both Vanilla Javascript (Javascript without abstractions) and Javascript Declarative Frameworks (specifically ReactJS). The term will begin with a heavy focus on HTML/CSS, and have topics of UI/UX, Accessibility, and Testing, trickled throughout. The course will be very heavy Vanilla JS to start while slowly tapering off, and will be very light on ReactJS at the start before rapidly getting into more depth.

The decision is to introduce ReactJS early (but lightly), instead of waiting until the middle of term, is to provide students with a very slow introduction to a challenging concept and really provide time to absorb ideas amidst these short 9 week teaching terms.

A visualisation of the teaching approach to topics can be seen below. The area of each shape denotes the focus it will be given with respect to time throughout the course.



3.3. What isn't included

There will be a number of people who feel strongly about the exclusion of particular topics from the course, such as typescript or more complex state managers (redux, mobx). Often when topics are omitted it's because they aren't *fundamental* knowledge in the limited weeks this course is offered in.

We've compiled an FAQ (https://gitlab.cse.unsw.edu.au/COMP6080/21T3/main-content/-/blob/master/FAQ.md) to answer these questions. If you still have further questions or comments, we'd encourage you to use the forum linked in the sidebar.

Students with prerequisite knowledge in this course should understand that this is an introductory course with very limited assumed knowledge, and as such, is not a course to extent an already firm foundation of knowledge. If you are already quite competent in the areas taught in this course, please be patient as we will be moving quite slowly by your standards. If this is something that you feel may be frustrating, it may be more appropriate with your skill-set to find a more stimulating course.

---

4. Teaching Strategies

This course uses the standard set of practice-focused teaching strategies employed by most CSE foundational courses:

- Lectures
- Tutorials
- Exercises
- Help Sessions
- Assignments
- Final Exam

4.1. Lectures

Lectures will be used to present the theory and practice of the techniques in this course. Although the lectures will primarily focus on the key concepts of software engineering, some lectures will also include practical demonstrations. Lecture slides will be available on the course web page.

On a typical week we will only have the live Thursday lecture (2 hours). The live Monday lecture will be replaced by approximately 2 hours of pre-recorded lectures to watch each week. Please note:

- It's critical that you watch the 2 hours of pre-recorded lectures each week *prior* to the Thursday lecture
- The length of pre-recorded and live lectures may vary week to week, but the intention is to keep the average weekly duration at 4 hours across the term.
- In general, industry driven lectures are more likely to be in the pre-recorded format
- Lecture notes for the live lectures may only be released a day or two before the lecture. This is because as part of the teaching strategy for this course the live lectures are designed to essentially "supplement" and "round-off" the pre-recorded lectures, so it's natural to wait for some feedback from students before over-planning these.

You can see the schedule for lectures here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60846) , and the list of lecture videos and content here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60838) .

Because lectures will be delivered from a *range* of people, it's important that students are prepared for differences in slide format and teaching style.

4.2 Tutorials

Tutorials are 1 hour lessons every week that help clarify ideas from lectures and work through exercises based on the lecture material. You should make sure that you use them effectively by examining in advance the material to be covered in each week's tutorial, by asking questions, by offering suggestions and by generally participating.

Tutorial information on it's release can be found by going to the tutorials page (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60837) . Tutorial solutions will be released at the end of the week the tutorial is in. Tutorials do not contribute to your final mark.

Tutorials will be run via Blackboard Collaborate.

4.3. Exercises

Exercises (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60837) for each week are provided for you to work through in your own time. These exercises are not marked or assessed in any way. You are not required to submit them anywhere, and tutors will not expect you to have completed them.

Solutions to these exercises will be released at the end of the week relevant to them (e.g. week 5 exercise solutions released at the end of week 5).

You can seek assistance from teaching staff to help you solve problems in exercises by attending help sessions.

4.4. Help Sessions

Help sessions are unprepared drop-in "clinics" where students and groups can go to seek help about course related matters, whether that be the project, tutorials, or exercises. Current tutors or lab assistants will supervise each help session.

Help session will be run via Blackboard Collaborate.

A timetable for help sessions can be found here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60836) .

4.5. Assignments

There will be a series of assignments which will run through the teaching period from weeks 1-10 and contributes to **75%** of the overall course mark.

- Assignment 1 is worth 15%
- Assignment 2 is worth 30%
- Assignment 3 is worth 30%

Assignments will be the platform you have to study and learn the material in substantially more depth. Your assignments will be completed via gitlab

The intention is that the third assignment will be completed in a pair. You are able to opt out of this and work alone (at your own expense!). You can choose your pair, otherwise we will assign you a partner. You can pair up with any student in the course.

Information on assignments can be found here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60835) .

4.6. Final Exam

There will be a centrally timetabled final exam which will in your UNSW exam timetable. The exam may contain a mixture of multiple choice questions, short answer questions, and programming exercises. More specific details of the exam will be provided through the course.

If you cannot attend the final exam because of illness or misadventure, then you must submit a Special Consideration request, with documentation, through MyUNSW within 72 hours of the start of the exam exam. If your request is reasonable, then you will be awarded a Supplementary Exam. No supplementary exams will be provided for students who score marks 49 or below on grounds of being "close" to a pass.
Information on the final exam can be found here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60834) .

---

5. Assessment

Item	Due	Weighting
Assignments	Due in weeks 4, 7, 10	75%
Final Exam	Exam period	25%

5.1. Assignments (75%)

The marking criteria for your assignments will be specified in each specification (once released) and can be found here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60835) .

5.2. Final Exam (25%)

The marking criteria for your final exam will be specified in the exam specification which can be (once released) found here (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60834) .

---

6. Course Schedule / Timetable

The schedule for this course is outlined clearly in the timetable for lectures (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60846) , tutorials/help sessions. (https://webcms3.cse.unsw.edu.au/COMP6080/21T3/resources/60836)

---

7. Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

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

---

8. Resources for Students

There is no single text book that covers all of the material in this course at the right level of detail and using the same technology base as we are. The lectures should provide sufficient detail to introduce topics, and you will then study them in further depth in the tutorials, exercises and assignments. For some lectures, further reading material may be given for students who wish to gain a deeper understanding.

Resource created 6 months ago (Saturday 15 May 2021, 11:21:20 PM), last modified 2 months ago (Tuesday 14 September 2021, 11:10:00 PM).

Comments (/COMP6080/21T3/forums/search?forum_choice=resource/60751) (/COMP6080/21T3/forums/resource/60751)



Alexander Bunn (/users/z5146667) 2 months ago (Mon Sep 13 2021 20:29:08 GMT+1000 (AEST))

Hey is the course title incorrect in the first table?