



UNSW

UNSW Course Outline

COMP6080 Web Front-End Programming - 2024

Published on the 12 Feb 2024

General Course Information

Course Code : COMP6080

Year : 2024

Term : Term 1

Teaching Period : T1

Is a multi-term course? : No

Faculty : Faculty of Engineering

Academic Unit : School of Computer Science and Engineering

Delivery Mode : Multimodal

Delivery Format : Standard

Delivery Location : Kensington

Campus : Sydney

Study Level : Postgraduate, Undergraduate

Units of Credit : 6

Useful Links

[Handbook Class Timetable](#)

Course Details & Outcomes

Course Description

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 will be given by current front-end developers from industry. These lectures primarily come from employees at Canva, a Sydney-based technology company that does a lot of work with front-end technologies.

COMP6080 is a challenging course. Front-end development is unlike most things you've experienced at university before. You will find the individual problems you solve much simpler than other level 6 courses, but the time you will feel that you spend on the aggregate of these issues will feel larger. A number of students will find this course quite time consuming if they're hoping to achieve a high mark. We'd encourage you to reflect on this fact before you enrol in the course.

Course Aims

COMP6080 aims to develop your confidence in building modern web-based applications to industry standards.

COMP6080 is a higher-level elective course and assumes that you have the following:

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

Postgraduate students may need to familiarise themselves with GIT, if not already familiar.

Course Learning Outcomes

Course Learning Outcomes
CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation
CLO3 : Use Javascript and CSS frameworks to allow more efficient integration of existing code and components into a final product
CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques
CLO5 : Design and build interfaces that focus on best user experience and accessible design practices

Course Learning Outcomes	Assessment Item
CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically	<ul style="list-style-type: none"> • Static front-end implementation • Dynamic front-end implementation • Web application implementation • Framework-based web application implementation • Final Exam
CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation	<ul style="list-style-type: none"> • Static front-end implementation • Dynamic front-end implementation • Web application implementation • Framework-based web application implementation • Final Exam
CLO3 : Use Javascript and CSS frameworks to allow more efficient integration of existing code and components into a final product	<ul style="list-style-type: none"> • Framework-based web application implementation • Final Exam
CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques	<ul style="list-style-type: none"> • Web application implementation • Framework-based web application implementation • Final Exam
CLO5 : Design and build interfaces that focus on best user experience and accessible design practices	<ul style="list-style-type: none"> • Framework-based web application implementation

Learning and Teaching Technologies

<https://cgi.cse.unsw.edu.au/~cs6080/23T3/course-outline>

Assessments

Assessment Structure

Assessment Item	Weight	Relevant Dates
Static front-end implementation Assessment Format: Individual	15%	Due Date: Week 3
Dynamic front-end implementation Assessment Format: Individual	5%	Due Date: Week 4
Web application implementation Assessment Format: Group	30%	Due Date: Week 7
Framework-based web application implementation Assessment Format: Group	30%	Due Date: Week 10
Final Exam Assessment Format: Individual	20%	Due Date: During Exam Period.

Assessment Details

Static front-end implementation

Assessment Overview

Students will build static front-end pages based on a reference image. Student's work will be marked by visual compliance checks, and for code quality checks. This criteria will be provided to both students and tutors. Tutors will share marks and comments with students.

Course Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation

Dynamic front-end implementation

Assessment Overview

Students will build a dynamic front-end page based on a form. Student's work will be marked by manual checking of their code's behaviour, and with code quality checks. This criteria will be provided to both students and tutors. Tutors will share marks and comments with students.

Course Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation

Web application implementation

Assessment Overview

Students will build a full web application front-end that works with a provided back-end server.

Student's work will be marked by manually checking for completed features; reviewing code to look for quality code; ensuring that the finished product is mobile responsive.

This criteria will be provided to both students and tutors. Tutors will share marks and comments with students.

Course Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation
- CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques

Framework-based web application implementation

Assessment Overview

Students will build a full web application front-end based on a modern framework that works

with a provided back-end server.

Student's work will be marked by manually checking for completed features; reviewing code to look for quality code; ensuring that the finished product is mobile responsive; ensuring that user experience and accessibility is expressed effectively; and ensuring that the application is well tested.

This criteria will be provided to both students and tutors. Tutors will share marks and comments with students.

For pair assignments, you complete them a pair of your choice. If you don't have a pair, we will find one for you. You can also choose to work alone, but we strongly do not recommend this as the workload is much higher for an individual.

Nominations for your pair (or to work alone) must be complete by the end of week 2. Information about this will be distributed in weekly notices.

Pairs will be **required** to contribute regularly to gitlab and in reasonably equal contributions as we still assess contributions individually (there is no blanket group mark assigned). Failure to do so may result in a loss of marks.

Course Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation
- CLO3 : Use Javascript and CSS frameworks to allow more efficient integration of existing code and components into a final product
- CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques
- CLO5 : Design and build interfaces that focus on best user experience and accessible design practices

Final Exam

Assessment Overview

There will be a centrally timetabled final exam which will appear in your UNSW exam timetable. The exam may contain a mixture of multiple choice questions, short answer questions, and programming exercises.

Course Learning Outcomes

- CLO1 : Able to apply Javascript semantics to design, construct, test and debug programs holistically
- CLO2 : Construct programs for web-front end with HTML, CSS, and DOM manipulation
- CLO3 : Use Javascript and CSS frameworks to allow more efficient integration of existing code and components into a final product
- CLO4 : Build stable applications that utilise concurrent programming through use of Javascript's asynchronous programming techniques

General Assessment Information

Grading Basis

Standard

Course Schedule

Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

General Schedule Information

The timetable for this course is outlined clearly in the timetable for [lectures](#), [tutorials](#), and [help sessions](#).

The content schedule is outlined below:

Week 1

- Course Overview
- Intro to Git
- HTML Fundamentals
- Image Types
- CSS Rules
- More Tags
- CSS Formatting
- CSS Layouts
- Flexbox
- SVGs
- Pre-processors
- zindex
- CSS Showcase
- Fonts

Week 2

- Demo: HTML/CSS Page
- Mobile CSS
- Using CSS Frameworks
- CSS Grids
- Dev Tools
- Javascript Intro (compared to C)
- Javascript Language Features & Syntax
- Javascript Advanced Functions

- The Javascript Ecosystem
- Node Package Manager
- NPM Advanced

Week 3

- Demo: Javascript - NodeJS
- Introduction
- DOM
- Events
- Javascript Closures
- Forms
- Local Storage

Week 4

- Demo: Javascript - Browser
- Events & Callbacks
- Promises
- Await / Async
- AJAX
- XMLHttpRequest (XHR)
- Fetch
- JSON & Data-interchange formats
- Understand HTTP Servers

Week 5

- Demo: Ass2 - Async & Planning
- UI Fundamentals
- Good & Bad UI 1
- Good & Bad UI 2
- Perceivability
- Operability
- Understandability
- Robustness

Week 7

- Using Git in pairs
- Introduction
- Basic "Global CSS" Usage
- Lifescycle
- Transpilation
- useState hook
- Demo: ReactJS Intro & A11y

- `useEffect` hook
- Using Fetch & Hooks
- Working with Multiple Files
- Components & Props
- Before you code

Week 8

- Demo: ReactJS in Ass3
- Linting
- Routing & SPAs
- CSS Frameworks
- `useContext`
- State management
- Class components
- Usability testing
- Figma & Component Libraries
- Pre-rendering

Week 9

- Demo: ReactJS Further
- Introduction
- Component Testing
- UI Testing
- UI Testing (Demo)

Week 10

- Final Exam Overview

Course Resources

Recommended Resources

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.

Course Evaluation and Development

This course is evaluated each session using the MyExperience system.

This is being addressed during 23T3.

- Assignments now explicitly link to the relevant lecture content to make it easier to follow
- Assignment 2 is now providing Static HTML which will allow us to both auto mark it and reduce student load
- Assignment 3 is now an individual assignment (as it was historically)
- Assignment 4 has a typescript template provided for those wishing to use it

Staff Details

Position	Name	Email	Location	Phone	Availability	Equitable Learning Services Contact	Primary Contact
Convenor	Hayden Smith					Yes	Yes

Other Useful Information

Academic Information

I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's [Special Consideration page](#).

II. Administrative matters and links

All students are expected to read and be familiar with UNSW guidelines and polices. In particular, students should be familiar with the following:

- [Attendance](#)
- [UNSW Email Address](#)
- [Special Consideration](#)
- [Exams](#)
- [Approved Calculators](#)
- [Academic Honesty and Plagiarism](#)

- [Equitable Learning Services](#)

III. Equity and diversity

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

IV. Professional Outcomes and Program Design

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: [https://www.unsw.edu.au/engineering/student-life/
student-resources/program-design](https://www.unsw.edu.au/engineering/student-life/student-resources/program-design).

Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.

Academic Honesty and Plagiarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW 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.*

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: student.unsw.edu.au/plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf

Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;
- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

Faculty-specific Information

[Engineering Student Support Services](#) – The Nucleus - enrolment, progression checks, clash

requests, course issues or program-related queries

[Engineering Industrial Training](#) – Industrial training questions

[UNSW Study Abroad](#) – study abroad student enquiries (for inbound students)

[UNSW Exchange](#) – student exchange enquiries (for inbound students)

[UNSW Future Students](#) – potential student enquiries e.g. admissions, fees, programs, credit transfer

Phone

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

School Contact Information

CSE Help! - on the Ground Floor of K17

- For assistance with coursework assessments.

The Nucleus Student Hub - <https://nucleus.unsw.edu.au/en/contact-us>

- Course enrolment queries.

Grievance Officer - grievance-officer@cse.unsw.edu.au

- If the course convenor gives an inadequate response to a query or when the course convenor does not respond to a query about assessment.

Student Reps - stureps@cse.unsw.edu.au

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)