

CPEN 321: Software Engineering (Summer 2022)

Contents

1. [Overview](#)
2. [Staff](#)
3. [Communication and Links](#)
4. [Project](#)
5. [Course Schedule](#)
6. [Grading](#)
7. [Absence](#)
8. [Acknowledgements](#)
9. [Academic Integrity](#)
10. [University Policies](#)

1. Overview

The course explores engineering principles for building non-trivial software-intensive systems: requirement specification, design, implementation, verification, and maintenance. The course uses a development project as the basic learning vehicle, providing students with an environment for applying the learned principles in practice while coping with constraints encountered when working on a real team: uncertain requirements, tight deadlines, etc. Students are evaluated based on their understanding of the software engineering principles and their ability to apply these principles in practice.

Learning Objectives

By the end of the course, students will learn several fundamental skills in modern software development:

- software engineering principles: requirements specification, design, implementation, verification, and maintenance of software;
- client-server application development paradigms, including mobile and cloud-based development;
- teamwork, inter- and intra-team communication;
- management of uncertainty in customer demands;
- ability to adapt quickly to new toolsets;
- efficient technical communication and presentation.

Course Format

- The class meets online two times a week for lectures (2x1.5h) and once a week for a lab section (2h).

- The grading is based on the development project, midterm, and participation marks. There is no final exam.
- Lectures are delivered synchronously on Zoom (not recorded).
- Labs are delivered synchronously on Discord (not recorded).
 - Facilitated by TAs
 - Provide the opportunity to ask clarification questions/get help with a particular milestone
 - Time for each team to meet and start to work on the next project milestone
- Attendance and active participation in lectures and labs is part of the grade.
- The mid-term exam will be held in one of the lectures.

Course Prerequisites

CPEN 221, CPSC 210, or equivalent.

2. Staff

Instructor	<u>Prof. Julia Rubin</u> Lectures: Mon. 5-6:30pm, Zoom Wed. 5-6:30pm, Zoom Office hours: Thu. 5:30-6:30pm, Zoom (or by appointment)
TAs	Khaled Ahmed Faridah Akinotcho Michael Tegegn Yingying Wang Labs: Tue. 5-7pm, Discord

3. Communication and Important Links

- Canvas is the main info hub: full syllabus, assignments, lecture slides, tutorials, grades, Zoom/Discord/Piazza links, etc.
- Lectures and office hours are held on Zoom
- Synchronous Lab meetings are held on Discord (**not monitored outside of lab hours!**)
- Discussions, Q/A, private messages to the course staff are on Piazza:
<https://piazza.com/ubc.ca/summer2022/cpen321softwareengineering>

Notes

- Discord and Piazza servers might be hosted outside of Canada. If you have issues using these services, please email the instructor immediately (by the end of the first week of classes)
- Grades for the assignments will be posted throughout the term. Students are encouraged to check it frequently and let the staff know of any mistakes and inconsistencies.

Support Resources

- [Student Guide to Canvas at UBC](#)
- [Student Guide to Zoom at UBC](#)
- [UBC COVID-19 Information for Students](#)
- [Managing Your Mental Health During the COVID-19 Outbreak](#)

Time Zone in Canvas

The course is set to Pacific Time (PT) time zone, where the University of British Columbia Vancouver Campus is located. All due dates are set to PT. Canvas will not automatically change time zones for you. If you want Canvas to display dates in your local time zone, you can go into your settings and adjust to your personal local time zone. Please refer to the Canvas guide on [how to set a time zone in your user account](#).

4. Project

A major part of the course is the development project. The project simulates real development settings, where development teams communicate with customers to implement software that satisfies the customers' requirements. The intent of this project is to give you the opportunity to apply the course concepts to a real system.

Scope: You are required to develop a client-server software system with the client (front-end) running on a mobile device and the back-end running on the cloud. Several examples of reasonably-scoped projects can be found [here](#).

Client side: a mobile application

- Must run natively on Android (not in a browser)
- Must run on at least one real device and on a Google Pixel 3 emulator running Android Q (API 29).
- Must be written in Android/Java
 - Using other frameworks, e.g., Expo, ReactNative, etc. is not permitted.

Server side: Node.js backend

- If your back-end uses a database, you may choose between MySQL and MongoDB. Using MongoDB Atlas/Realm is not allowed.
- Must run in a cloud (AWS, Azure, Google, IBM, etc.)

- Most vendors, such as Amazon, Microsoft, Google, and IBM, provide free student subscriptions.
- The course staff can also provide you with a pre-paid subscription to Microsoft Azure. If interested, send a request to the instructor on Piazza with (a) your full name, (b) your project name, and (c) your Microsoft account id, if available, or an email address that will become your account id.

Code Management: All development artifacts must be stored in Git - a popular version control system.

- All code of each project (front-end and back-end) should be stored in one Git repository
 - Well-structured internally
 - (<https://www.wired.com/2015/09/google-2-billion-lines-codeand-one-place/>)
- You must give all TAs access to your repository
- Feel free to create a throw-away/anonymous GitHub or Bitbucket account for this course.
- Use the [Atlassian Git Introduction](#) can get you started with Git. A shorter, less formal, [guide](#) is also available.

Milestones and Deliverables

- M1: Android App (individual)
- M2: Android App with Node.js Backend (individual)
- M3: Requirements (group)
- M4: Design (group)
- M5: MVP (group)
- M6: Test Design and Setup (group)
- M7: Code Review (group)
- M8: Testing (group)
- M9: Final Release (group)

The detailed specification for each milestone deliverable is attached to the Canvas assignment. The milestone submission schedule is also outlined in the [Course Schedule table](#).

Project Groups

- Milestones M1 and M2 are performed individually.
- Milestones M3-M9 are performed in groups of 4 students (2 students for the front-end and 2 for the back-end).
- Use Piazza “Search for Teammates” page to find team members.
- Choose your team members wisely as they contribute to the success of your team; no changes will be permitted.
- If you do not have a team organized by the **Monday June 6, 11pm PST** deadline, we will assign you to a team randomly.
- We will “fill up” all teams to have 4 members, potentially “breaking” teams of 3 or less to multiple teams

Late Deliverables

- Each student is given 2 grace days per term in total that can be used for any individual milestone (M1 and M2).
- Each group is given 6 grace days per term in total that can be used for any group milestones (M3-M8).
- You can use the grace days as you wish i.e., all three days on one deliverable, distribute across deliverables, etc.
- Grace days are counted as integers, i.e., if you are one hour late, that counts as one grace day.
- Once all grace days are used, late submissions will receive the grade of 0.
 - In exceptional circumstances, apply for AppSci's academic concession.

5. Course Schedule

Note that this is a tentative week-by-week guide and thus is subject to change.

	Date	Topic	Major Deadlines (by Thursday of that week, 11pm PST, unless stated otherwise)
W1	Mon, May 16, 2022	Introductions, Goals, and Logistics	
	Wed, May 18, 2022	Software Lifecycle, Development Processes	
W2	Mon, May 23, 2022	<i>Victoria Day</i>	
	Wed, May 25, 2022	UML	M1: Android App (individual, 5%)
W3	Mon, May 30, 2022	Requirements	

	Wed, June 1, 2022	Requirements	M2: Android App with Node.js Backend (individual, 8%)
W4	Mon, June 6, 2022	Architecture, Design	Groups formed and submitted on Canvas - <i>by Monday June 6, 11pm</i>
	Wed, June 8, 2022	REST, Microservices	M3: Requirements (group, 5%)
W5	Mon, June 13, 2022	<i>Project presentations</i>	
	Wed, June 15, 2022	<i>Project presentations</i>	
W6	Mon, June 20, 2022	Teamwork, Advanced Version Control	M4: Design (group, 5%) - <i>by Monday June 20, 11pm</i>
	Wed, June 22, 2022	Code Reviews, Anti-patterns	
W7	Mon, June 27, 2022	<i>Exam Break</i>	
	Wed, June 29, 2022	<i>Exam Break</i>	
W8	Mon, July 4, 2022	Testing	
	Wed, July 6, 2022	Testing	M5: MVP (group, 12%)

W9	Mon, July 11, 2022	Verification and Validation, Analysis	
	Wed, July 13, 2022	Verification and Validation, Analysis	M6: Test design and setup (group, 5%)
W10	Mon, July 18, 2022	<i>Preparation for mid-term</i>	
	Wed, July 20, 2022	<i>Preparation for mid-term</i>	M7: Code review (group, 5%)
W11	Mon, July 25, 2022	<i>Mid-term</i>	
	Wed, July 27, 2022	<i>Recap, Q/A, and preparation for final project presentation</i>	M8: Testing (group, 5%)
W12	Mon, Aug 1, 2022	<i>BC Day</i>	
	Wed, Aug 3, 2022	Continuous Integration, DevOps	M9: Final release (group, 15%)
W13	Mon, Aug 8, 2022	<i>Open to topics of interest or guest lecture</i>	
	Wed, Aug 10, 2022	Top project presentations and awards	

6. Grading

Grading for this course is based on four components:

- **Project** – 65%
 - M1: Android App (individual, 5%)

- M2: Android App with Node.js Backend (individual, 8%)
- M3-M4, M6-M8: 5% each (group, 25% in total)
- M5 (MVP): group, 12%
- M9 (Final Release): group 15% (assessed based on both the quality of the final deliverable and the quality of the SE process followed)
- **Mid-term** – 20%
- **Participation** – 15%
 - Peer-evaluation by team members – 5%
 - Active participation in class – 4 %
 - Lab participation / TA evaluation – 4%
 - Helping to answer piazza questions, etc. – 2%

7. Absence

Attending lectures, labs, and the mid-term exam is mandatory. In special circumstances, you should submit a request for academic concession [here](#).

8. Acknowledgements

We gratefully acknowledge the inspiration we drew and part of the material that we adapted from the previous editions of CPEN 321 and CPSC 310 (University of British Columbia), as well as from CSE 403 (University of Washington), CSC C01 (University of Toronto), CS 169 (UC Berkley), and CS428 (University of Illinois at Urbana-Champaign).

9. Academic Integrity

The academic enterprise is founded on honesty, civility, and integrity. As members of this enterprise, all students are expected to know, understand, and follow the codes of conduct regarding academic integrity. At the most basic level, this means submitting only original work done by you and acknowledging all sources of information or ideas and attributing them to others as required. This also means you should not cheat, copy, or mislead others about what is your work. Violations of academic integrity (i.e., misconduct) lead to the breakdown of the academic enterprise, and therefore serious consequences arise and harsh sanctions are imposed. For example, incidences of plagiarism or cheating may result in a mark of zero on the assignment or exam and more serious consequences may apply if the matter is referred to the President's Advisory Committee on Student Discipline. Careful records are kept in order to monitor and prevent recurrences.

For more information, see [here](#).

10. University Policies

UBC provides resources to support student learning and to maintain healthy lifestyles but recognizes that sometimes crises arise and so there are additional resources to access including those for survivors of sexual violence. UBC values respect for the person and ideas of all members of the academic community. Harassment and discrimination are not tolerated nor is suppression of academic freedom. UBC provides appropriate accommodation for students with disabilities and for religious, spiritual and cultural observances. UBC values academic honesty and students are expected to acknowledge the ideas generated by others and to uphold the highest academic standards in all of their actions. Details of the policies and how to access support are available [here](#).