# Software Construction II
## CPEN 320

## COURSE INFORMATION

| | |
|---|---|
| Instructor: | Ali Mesbah |
| Homepage: | `https://people.ece.ubc.ca/amesbah/` |
| E-mail: | `amesbah@ece.ubc.ca` |
| Office Hours: | by appointment |
| Course Web page: | UBC Canvas |
| Discussion forum: | Piazza<br>`https://piazza.com/ubc.ca/winterterm12024/cpen320` |
| Teaching Assistant Info: | |
| | Nashid Noor |
| | Taha Shabani |

You can contact the instructor and TAs through Piazza (publicly or privately).

### COMMUNICATION

Rather than emailing the teaching staff, students are encouraged to post questions on the discussion forum. Active and constructive participation on the forum counts towards participation marks. Find our Piazza page (see above).

## LECTURE DATES, TIMES, ROOMS

| | |
|---|---|
| Lecture: | Mon & Wed, 14:00–15:30 |
| Lab: | See the UBC Course Calendar for your lab section |

## COURSE DESCRIPTION

In the rapidly evolving field of software engineering, the "Software Construction: II" course stands as a beacon for third-year undergraduate students aiming to learn contemporary software construction concepts and techniques. This course is designed to provide an in-depth understanding and hands-on experience in developing ubiquitous and intelligent software applications.

Students will dive deep into the core aspects of modern software construction, starting from understanding distributed (e.g., client/server and peer-to-peer architectures) progressing through the intricacies of web protocols and standards, and mastering the art of asynchronous programming and REST APIs. A focus will be on learning and applying cutting-edge AI technologies to enhance the functionality and user experience of applications.

By the end of the course, students will be proficient in designing and implementing dynamic and responsive web applications that intelligently leverage AI capabilities. They will be adept at integrating various AI models and ensuring web security, scalability, and performance. The course culminates in hands-on assignments where students will gain practical exposure to server-side programming using popular languages and frameworks and will understand how to employ various types of AI models to solve real-world problems and will have the opportunity to bring their innovative ideas to life by creating a web application enriched with intelligent features.

# LEARNING OBJECTIVES

Upon successful completion of the course, students will be able to:

- **Understand Architectural Patterns:**
  - Differentiate between client/server and peer-to-peer architectures.
  - Describe the advantages and use cases of different network architectures.
- **Web Data Protocols and Content Delivery Networks (CDNs):**
  - Understand and apply web data protocols such as HTTP, WebSockets, and RPC in web development.
  - Explain the roles and functionalities of these protocols in enabling web communications.
  - Understand the purpose and advantages of using CDNs in distributing web content.
- **Web Page Development:**
  - Develop web pages using HTML, CSS, JavaScript, and responsive design principles.
  - Understand the differences and appropriate use cases for static HTML pages vs. dynamic DOM.
- **Asynchronous Programming:**
  - Comprehend asynchronous programming concepts and apply them in building more responsive web applications.
- **Stateless vs stateful programming:**
  - Understand how server-side code interacts with client requests.
  - Understand the concepts behind stateful and stateless communication.
  - Leverage RESTful APIs and understand their role in scalable software development and integration.
  - Manage user sessions and data effectively using cookies, session management techniques, and various data storage options.
- **Software as a Service (SaaS) and Continuous Delivery:**
  - Understand the architecture and delivery models of SaaS including serverless and monolithic approaches.
- **Security, Concurrency, Scalability, and Performance:**
  - Understand common security threats and mitigation techniques.
  - Be familiar with authentication and authorization mechanisms and understand their importance in securing web applications.
  - Understand the concepts of concurrency.
  - Apply techniques for improving the scalability and performance of web services such as load-balancing.
- **Testing, Accessibility, and Internationalization:**

  - Apply software analysis and testing techniques for quality control.
  - Ensure that web applications are accessible to a diverse user base and adhere to internationalization standards.

- **AI Integration:**
  - Understand the potential and limitations of applying AI in software construction.
  - Become familiar with AI concepts such as Deep Neural Networks, Foundation Models, Large Language Models, and Prompt Engineering, and their significance in AI-driven software engineering.
  - Integrate basic AI functionalities into software applications.

## COURSE ORGANIZATION / STRUCTURE

This course is structured to engage students through a variety of learning activities designed to provide a deep understanding and practical experience in software verification and testing. The course will be organized as follows to meet the stated learning objectives:

- Lectures: Regular lectures will cover the theoretical foundations, principles, and emerging trends in software testing and verification.
- Lab Assignments: students will be given assignments to gain hands-on practical experience. This will encourage the application of theoretical knowledge to real-world problems and foster teamwork and problem-solving skills.
- Report Writing and Presentations: Students will document their work in well-constructed reports, enhancing their technical writing skills; they will also present their lab work enhancing their communication skills.
- Examinations: Comprehensive examinations (e.g., midterm, final) will be conducted to assess understanding of the theoretical concepts and the ability to apply them to practical scenarios.

## STUDENT EVALUATION

| | |
|---|---|
| Active Participation: | 05% |
| Midterm (**October 16**): | 20% |
| Assignments: | 35% |
| Final Exam: | 40% |

Active participation includes class attendance, participation in discussions in the class and on Piazza, participation in in-class exercises, asking good questions, good presentations (if applicable).

## COURSE MATERIALS

Course material includes lecture notes, and shared papers/articles. Lecture notes will be made available through Canvas.
The following books and resources are recommended:

- **Engineering Software as a Service**, by Armando Fox & David Patterson. Free PDF: `https://drive.google.com/file/d/19dqWms9qUQ2Bi8JfsBhC5Q6zDeJd0s5o/view`

- **Building Intelligent Systems: A Guide to Machine Learning Engineering**, by Goeff Hulten. UBC Library provides access to an online version of the book.
- Web technologies:
  - **Getting Started with the Web** `https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web`
  - **Eloquent JavaScript**, by Marijn Haverbeke. Free PDF: `https://eloquentjavascript.net/Eloquent_JavaScript.pdf`

# PREREQUISITES

Pre-reqs: CPEN 221, CPSC 221.

# Policies

## General Policies

- Assignments will be done in groups of two. You need to pick a partner by the beginning of the second week. You are free to discuss, share and collaborate with your partner without reservations. However, both of you will be jointly responsible for the solution you turn in. You may not share code or discuss ideas with any other pair of students working on the assignment. Further, both you and your partner will be awarded the same grade for the assignment. You are welcome to split up the work anyway you want. However, both of you will be expected to know the details of your solution to each assignment. We reserve the right to call upon you individually to explain the details of your solution. Failure to do so will result in you alone earning a 0 for the assignment.

- The assignments cumulatively build upon each other, so not turning in even a single assignment can adversely impact your grade for future assignments. You will need to turn in the assignment in the lab on which it is due, by physically coming to your lab session. If you cannot come to the lab to turn in the assignment, you must make arrangements with TA's well ahead of time. Otherwise, you will receive a 0 for the assignment.

- All deadlines are hard unless you have a documented emergency. You may be called upon to produce documentation related to the nature of the emergency.

- Finally, it is your responsibility to keep up with course announcements, lectures and assignments via Canvas, Piazza, and coming to the lectures.

## USE OF AI

In recognition of the evolving landscape of software development and the increasing role of artificial intelligence (AI) in coding, this course allows the exploration and utilization of AI tools to enhance learning and productivity. However, all students must adhere to the following guidelines to ensure ethical use and integrity in learning:

1. **Authorized Use:** Students may use AI tools such as CoPilot or ChatGPT, for code generation, debugging, or optimization as part of their assignments given they meet the conditions listed here.

2. **Understanding and Originality:** While AI can assist in generating code or providing solutions, students are expected to understand and be able to explain any code or content submitted. Reliance on AI should not replace the student's own learning, problem-solving, and coding skills.

3. **Attribution:** Any use of AI-generated code, text, images, or insights must be fully and clearly attributed, specifying what portions of the work were assisted by AI; the disclosure must be clear in their reports, code, presentations, etc. For code, explicit code comments are needed indicating the nature of the AI's contribution.

4. **Ethical Considerations:** Students must use AI tools responsibly, considering the ethical implications and avoiding any form of misuse that could lead to dishonesty, plagiarism, or other forms of academic misconduct.

5. **Enhancing Learning, Not Replacing It:** The intent of allowing AI in code assignments is to enhance educational outcomes, foster innovation, and familiarize students with tools they might use in their professional lives. It should not diminish the learning process or the acquisition of fundamental coding skills.

6. **Instructor's Discretion:** The instructor reserves the right to limit or prohibit the use of AI tools in any assignment if deemed necessary to preserve academic integrity or meet specific learning objectives.

Violations will be referred to the dean and may result in receiving a 0 in the course and suspension.

## UBC ACADEMIC HONESTY AND STANDARDS

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 UBC 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, incidents of plagiarism or cheating may result in a mark of zero on an 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: `http://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,286,0,0`

## NON-ACADEMIC MISCONDUCT

Mistreatment towards anyone in our Engineering community is not acceptable. Mistreatment is disrespectful or unprofessional behavior that has a negative effect on you or your learning environment, or conduct that is contrary to the principles that support a respectful environment. This includes making demeaning, offensive, belittling, and disrespectful comments, using abusive language, engaging in bullying, harassment, and discrimination. If you have witnessed or been subject to mistreatment, there are people and support resources here to help. Find out how to get support or discuss an issue related to discrimination, bullying, harassment, or sexual misconduct through the non-academic misconduct link below: `https://academicservices.engineering.ubc.ca/degree-planning/non-academic-misconduct-discrimination-and-edi-i-support/` ECE students, faculty, and staff are also welcome to submit comments, suggestions, and requests around Equity, Diversity, Including and Indigeneity (EDII) in the ECE Department to our EDII Suggestion Box.

Submissions can be anonymous, and are received directly by the ECE EDII Committee for review: `https://ece.ubc.ca/engage-with-ece/edii-suggestion-box/`

## HEALTH AND WELLNESS

UBC provides resources to support student learning and to maintain healthy lifestyles, while recognizing that challenges and crises can arise for students. There are resources in ECE and at UBC where students can find can help and support, including wellness, equity, inclusion and indigineity, resources for survivors of sexual violence, and health. Some frequently used resources are as follows:
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 UBC's respectful environment policies, which all students, staff and faculty are expected to follow, can be found here: `https://hr.ubc.ca/working-ubc/respectful-environment`

## ACADEMIC CONCESSION

The University is committed to supporting students in their academic pursuits. Students may request academic concession in circumstances that may adversely affect their attendance or performance in a course or program. Students who intend to, or who as a result of circumstance must, request academic concession must notify their instructor, dean, or director as specified in the link below. `https://www.calendar.ubc.ca/vancouver/index.cfm?tree=3,329,0,0` Students seeking academic concession due to absence from the final exam for any reason must apply to Engineering Academic Services (EAS) within 72 hours of the missed exam. This is a standard practice for all final examinations at UBC. For more information, see: `https://academicservices.engineering.ubc.ca/exams-grades/academic-concession/`

## LAND ACKNOWLEDGMENT

This course is held on the UBC Point Grey (Vancouver) campus, which sits on the traditional, ancestral, unceded territory of the the Coast Salish Peoples. UBC is implementing its Indigenous Strategic Plan, taking a leading role in the advancement of Indigenous peoples' human rights. To learn more about the Faculty of Applied Science's role in building upon the Indigenous Strategic Plan and committing to Truth and Reconciliation, please visit: `https://apsc.ubc.ca/EDI.I`