Lecture 0: Introduction to the Course

CPEN322 - Building Modern Web Applications - Term 1 2021

Karthik Pattabiraman

The Univerity of British Columbia
Department of Electrical and Computer Engineering
Vancouver, Canada

Electrical and Computer Engineering

September 9th, 2021

Instructor: Karthik Pattabiraman

- Professor at UBC, ECE
  - PhD from UIUC (2008)
  - Post-doc at Microsoft Research (2009)
  - Faculty member at UBC (since 2010)
  - Sixth time I'm offerring this course - also created it
- Research
  - Internet Of Things (IoT)
  - Security and Reliability
  - Error Resilient Systems
  - Software Engineering

Course TAs:

- Pritam Dash
  - PhD student
- Kumseok Jung
  - PhD student
- Abraham Chan
  - PhD student
- Ali Asgari
  - Masters student

For any question about the assignments, please ask the TAs.

- Questions about assignments should be asked on Piazza only
- For private matters, write a private message to "Instructors" on Piazza.
- No questions will be answered **48 hours** before deadline
- **No Email whatsoever - we'll ignore all email messages**

What's this course about?

- **Core principles** behind building modern web applications
- Abstractions and design principles
- Application of technologies such as CSS, HTML, JavaScript, node.js to the above

What's it not about ?

- Learning of specific technologies
  - These will most likely get outdated by the time you finish
  - Fast changing field, so new technologies tommorrow
  - Can learn any technology if you understand the principles and concepts behind web development
- Frameworks or libraries (e.g., React)
  - These are built on the principles and concepts
  - Too many to cover in a reasonable time

Why take this course ?

- You will understand the **principles** behind web application development
  - Not simply copy-paste code from websites to string together a web application
  - You will understand **why** technologies are the way they are, rather than accept it as a statement of fact, and perhaps change them if needed
  - It enables you to design novel techniques and technologies in the web application space
  - If you put in the effort, this course will be really fun! :-)

Why not to take this course ?

- You just want to write a (lot of) web code
  - Online tutorials will teach you how to do this
  - While you'll do a series of programming assignments, their focus is to teach you the principles
- You want to impress your future employer with cool-sounding buzzwords
  - There won't be many of these unfortunately
- You want an **easy** final year elective course
  - This course will require significant work. It will not be easy.

Pre-requisites

- EECE 210 or equivalent (e.g., CPSC 210)
  - Principles of software development
  - Knowledge of invariants, specifications etc.
  - Experience using at least one OOP language (e.g., Java)
- Maturity to tackle large software development tasks
- No Web programming/JavaScript experience is needed
  - However, you should be able to pick it up quickly
  - Invest considerable time **outside** of class in learning JavaScript

Grading

- Assignments (40%): Five programming assignments worth 8% each, building on each other.
- Exams and quizzes (50%): One Midterm (13%) and a final exam (25%); Quizzes (12%)
- Class participation (5%): Asking and answering questions on Piazza. NOTE: Quality more important than quantity.
- JavaScript Proficiency quiz (5%): Test you on the basics of JavaScript (online)

Lectures

- Delivered by Karthik as asynchronous videos - released every week on Tuesdays
- Weekly quizzes will test knowledge of material in the videos (due most Mondays)
- Problem solving sessions held every Tuesday and Thursday during class time (**in person**)
- Attendance is non-mandatory at synchronous sessions; but exam questions will be similar to these
- Will release both the problems and their solution on Github and Piazza, but no recordings of sessions
- You'll attend only one of the Tuesday or Thursday sessions each week depending on your student no.

Reference Books (non-mandatory)

1. "Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke
2. "JavaScript: The Good Parts" by Douglas Crockford (highly recommended)
3. "Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Moderns JS Libraries" by Eric Elliott
4. "Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript" David Herman
5. "JavaScript: The Definitive Guide" by David Flanagan
6. "Secrets of the JavaScript Ninja" by John Resig, Second Edition (highly recommended)

Assignments

- Five Assignments where you'll build a complete web application from scratch (i.e., chat server and client)
  - Assignments build cumulatively on each other. Missing even one lab means you'll lose big !
  - Solutions will not be provided for any of the assignments
  - To be done individually; No collaboration allowed (even for ideas)
  - Submission via Github (later) - physical presence not needed

Exams & Quizzes

- One mid-term exam and one final exam (cumulative). Distribution is as follows:
  - Midterm: 13% (November 2nd)
  - Final exam: 25% (in December)
  - Quizzes every week: 12%

Exams and Quizzes are open-book, open notes. But collaboration is strictly prohibited.

Piazza and Class Participation

- We will use Piazza for all course-related communication
  - Do NOT email teaching staff (unless it's an emergency)
  - Use private posts for personal situations
  - Use appropriate tags for your posts
- You will receive class participation points for asking good questions and answering questions on Piazza
- **Class participation**: Counts for 5% of the course grade - based on Piazza alone
- Complete survey of Piazza names on Canvas by Sept 21st - graded for 1 point

Lab Information

- Labs are NOT needed, as all assignment submission and grading will be done offline
- You need to register for a Github account if you don't have one already (survey on Canvas)
- Lab times may be used by us to ask you questions on the assignment submissions
- All questions about assignments must be asked on Piazza, NOT in Karthik's office hours
- Labs are **the only way to get one-on-one assignment help** from the TAs (in person)

Programming Proficiency Quiz

- Will test you on basics of JavaScript (Sep 21st): Self-study
  - 5 programming problems of 1 mark each - no partial credit will be given
  - Needs to pass provided **all** test cases for each program. No partial credits.
  - To be taken during the class time on Sep 21st on HackerRank
  - Start preparing for the test now. Will release practice test by today.
  - We don't care about code quality. Instantaneous feedback.

Web Developer Tools & Editor

- Your favorite web browser + built-in web dev tools
  - Chrome with DevTools
  - Firebug (also include some great web development tools!)
- Node.js (latest version)
- The text editor of your choice :-)
  - **VsCode (recommended)**
  - Atom, Sublime etc.
  - Notepad++
  - Vim/Emacs
- IDEs can be used for Web Development

Additional Tools to be installed

- Git client
- (Optional): GitHub Desktop Client for Windows/Mac
- Npm (later in the course)
- MongoDB (later in the course)

Final Thoughts

- Do you really want to take this course ?
  - Involves significant amount of work and time
  - Easier electives are available in your final year
  - Requires
- If you're staying, welcome on board !
  - This is the seventh time this course is being offered (sixth time I'm offerring it)
  - Significant changes this time based on feedback from previous years
  - So please feel free to give us suggestions for improvement (these are actively encouraged)
  - Tell us what you liked and what you didn't like - we've incorporated past suggestions.

To do by Next Week

Immediate Action Items

- Complete Piazza survey on Canvas
- Watch the recorded video and take the first quiz
- Create Github account and share username with us on Canvas

- If you are not familiar with JavaScript yet, start learning it asap
  - Programming proficiency quiz on Sep 21st
  - We'll post a sample quiz and solutions by later today