Lecture 0: Introduction to the Course

CPEN400A - Building Modern Web Applications - Winter 2018-1

Karthik Pattabiraman, and Julien Gascon-Samson

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

Electrical and Computer Engineering

Tuesday, September 6th, 2018

Instructor: Karthik Pattabiraman

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

Course TAs:

- Kumseok Jung
  - Second year Masters student
- Aarti Kashyap
  - First year Masters Student
- Pritam Dash
  - First year Masters student

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

- You should prioritize attending and asking questions during lab sessions (later)
- Alternatively, you are encouraged to ask your questions on Piazza publicly.
- For private matters, write a private message to your TA on Piazza.
- **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  - These will get outdated by the time you finish
  - Fast changing field, so new technologies tomm.
  - Can learn any technology if you understand the principles and concepts behind web development
- Frameworks or libraries (e.g., Jquery)
  - 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 lab assignments worth 8% each
- Exams (50%): One Midterm and a final
- Class participation (5%): In-class participation and for asking and answering questions on Piazza. NOTE: Showing up is neither necessary nor sufficient for participation.
- JavaScript Proficiency quiz (5%): Test you on the basics of JavaScript (MUST PASS)

Lectures

- Delivered by Karthik
- Will consist of a mix of teaching (lecturing) sessions mixed with in-class activities
  - Please bring your laptops fully charged with you to class as it will be easier for in-class activities
  - You will work in teams of 2 or 3 in class on the activities
  - Participation in class activities is important as exam questions will be similar
- Lecture notes will be distributed ahead of time – no course textbook required
  - However, you should keep your own notes

Reference Books (non-mandatory)

1. "Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke
2. "JavaScript: The Good Parts" by Douglas Crockford (where JavaScript quiz is from)
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. "You Don't Know JS" by Kyle Simpson

Assignments

- Five Assignments where you'll build a complete web application from scratch
  - Assignments build cumulatively on each other. Missing even one lab means you'll lose big !
  - To be done in teams of two (Choose partner by next week in the same lab session as you)
  - Attendance compulsory at lab sessions when the assignments are due (optional in other sessions)

Exams

- One mid-term exam and a final exam (cumulative). Distribution is as follows:
  - Midterm: 20% (in October)
  - Final exam: 30% (in December)
- **Need to pass the exams cumulatively in order to pass the course**
  - Need $> 25/50$ in the 2 exams combined together

Programming Proficiency Quiz

- Will test you on basics of JavaScript (Sep 18th): Self-study
  - 5 programming problems of 1 mark each - no partial credit will be given
  - Needs to pass provided test cases for each program. Instantaneous feedback.
- **Need to pass the programming proficiency quiz in order to pass the course**!
  - Need to get at least 3 of 5 questions correct

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 specific situations
  - Use appropriate tags for your posts
- You will receive bonus class participation points for asking good questions and answering questions, both in Piazza and in-class
- **Class participation**: Showing up is a necessary but NOT a sufficient condition for participation

Web Developer Tools & Editor

- Your favorite web browser + built-in web dev tools
  - Chrome with DevTools
  - Firebug (also include some great web development tools!)
- The text editor of your choice :-)
  - Sublime
  - Atom
  - Notepad++
  - Vi/Emacs
- IDEs can be used for Web Development

Additional Tools to be installed

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

Lab Information

- Labs begin in week #2. First lab:
  - L1A: September 12 (Wednesday)
  - L1B: September 14 (Friday)
- **Attendance mandatory in first lab** to make sure you are all set to work on your assignments and to meet with your lab mate and TA.
- You should have already formed a team of two at this point and sent your team info to the TAs (see next slide).
- Subsequent labs labs are only mandatory on the week of assignment submission - both team members should attend
  - But you are highly encouraged to attend other sessions as they serve as TAs' office hours. No other office hours will be held by the TAs.
  - Need to start early if you want help during the lab sessions - we'll not provide help in labs when assignments are due

Git

- Open source distributed version control system
- We will be using Git for version control and GitHub for hosting
- Each group will receive a private GitHub repository with us as collaborators
- Once you find a partner in the same lab, send a private Piazza post to your TA with the names of your group + lab section + GitHub usernames

Assignment Submission

- Assignment submissions will take place through GitHub
- Create an assignment branch (i.e., assignment-1, assignment-2, assignment-3, assignment-4, assignment-5) by the due date
- No late commits will be accepted (unless with instructor permission or in the case of documented emergencies).
  - **Please push your latest changes to the appropriate branch before 11:59:59 PM on the due date!**
  - **Make sure the assignment branch is working prior to the deadline. You can continue development on main branch.**

Git Demo

1. Clone repository
2. Commiting changes
3. Pushing/pulling changes from repository
4. Branching

Useful Git Commands

- git clone
- git pull origin master
- git push origin master

Creating Branches

- git branch assignment-X
- git checkout assignment-X
- git push -u origin assignment-X
- git checkout master
- git branch
- git branch -r

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
- If you're staying, welcome on board !
  - This is the fourth time this course is being offered (third time I'm offerring it)
  - 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

Find a partner to do the assignments with

- Must be in the same lab session as you–no exceptions
- Ask them whether they're going to drop the course
- Both you and your partner get the same marks
- If you break up with your partner at any point, you need to do the work alone for future assignments (both of you'll get the common code)
- Once you find a partner, one of you should send a private Piazza post to TAs, indicating your names, lab section, and GitHub usernames

- If you are not familiar with JavaScript yet, start learning it by yourself (fast)
  - Programming proficiency quiz - must pass to pass the course
  - We'll post a sample quiz and lecture materials online by today.