EEP 520- Software Engineering for Embedded Applications

Week 1

Teaching Team

- **Instructor:**
  - **Sep Makhsous,** sosper30@uw.edu
    - Office hours: Sign up using the Calendly (see Canvas).
- **TA:**
  - **Sanskar Naik** snaik9@uw.edu
- **Grader**
  - **Brook Hu** bukenhu@uw.edu

- **Lab hours (ECE room 361): CLICK HERE**

Class Time and Location

- Class Meetings Information:
- Lecture:
  - MW 4:00p - 5:50p. ECE 045
- We will be holding in person lectures, your attendance is VERY important; I will be recording everything via zoom but keep in mind this is not a hybrid course and the interactions via zoom will be very limited. Come to class, engage, and do the exercises with your classmates in real time and I will guarantee your success in this class.

Course Structure

- Lecture Format:
  - We have two 1 hour and 50min sessions each week
    - You will get prefilled slides for the entire week by 12 hours before Monday lecture
    - Filled out slides will be uploaded within 12 hours of the lecture
    - All lectures will be offered via zoom and will be recorded
    - Attendance is HIGHLY recommended
  - Each week you will have 1 ICTE, dates will not be announced
    - Most ICTEs must be done during the class unless specified by the teaching team
    - ICTEs are done in teams of 2 to 3

Course Structure

- Homework:
  - See the Canvas home page for homework assignments and deadlines.
  - **Workload**: There will be approximately six (6) homework assignments.
  - HW are an integral portion of class learning. Failure to make a good-faith effort at the hw is grounds for failing the class.
  - Individual Submission
  - Due on Friday at 11:59 PM (for 5% bonus)
    - Late submissions accepted until Saturday at 11:59 PM – Full credit but no bonus
- Final Project
  - Your final project will be done individually.
- Bonus
  - There will be 5 to 6 bonus assignments through out the quarter

Grading

- Homework: 50%
- ICTEs: 20%
- Final Project: 30%
- Bonus: 5%
- **Total: 105%**

Textbook

There is no required textbook for this course, i.e. you do not need to purchase a textbook. Lecture notes will be provided on canvas

I will draw material from several references:

- The git book. https://git-scm.com/book/en/v2
- Kerninghan and Ritchie, C Programming, 2nd Edition, https://github.com/hiverkiya/Dennis-Ritchie-C-Solutions/blob/master/the_c_programming_language_2.pdf
- GNU Make. https://www.gnu.org/software/make/manual/make.html#Introduction
- Books about C++: https://stackoverflow.com/questions/388242/the-definitive-c-book-guide-and-list

Teamwork

Done in teams: ICTEs - Groups of 2-3

Done Individually: HW, Bonus, and Final Project

In addition, you are welcome (and encouraged) to:

- work together, synchronously and asynchronously, in study groups;
- use analytical and numerical computational tools -- specify the tool(s) in source code and/or text;
- reuse example source code and other materials provided in this course;
- consult textbooks, websites, and other publicly-available materials -- include a full citation(s) with the URL and/or DOI (Links to an external site.).

Submission guidelines

You will submit your homework writeup by uploading a .pdf on the Canvas Assignment. We will only grade legible .pdf files -- we will not grade content in any other file format (.doc, .zip, .ipynb, .m, ...).

If you write your solutions by hand, you must create a legible scan; if you have any doubts about the fidelity of your scans, send a sample to the instruction team in advance of the homework deadline.

Communication

We will use Canvas (i.e. this site) extensively for course materials and **Slack** for all communications.

The instruction team will provide homework, example code, etc. through Canvas; you will submit homework electronically through Canvas as described above.

If you have a question -- about a concept, HW problem, etc. -- it's likely someone else in the class does as well. Please use our Slack channel to post questions (rather than emailing or messaging the instruction team directly) so that (a) others get to propose answers and (b) others get to see the definitive answer (if any). If you send questions via email to the instruction team, we will direct you to ask it on Slack so others can benefit from our answers.

What we will (try to) cover

- Week 1: Course overview. Introduction and build environment.
- Week 2: Introduction to C programming.
- Week 3: C programming and Abstract Data Type (ADT)
- Week 4: Introduction to C++.
- Week 5: C++ and Standard Template Library (STL)
- Week 6: STL II
- Week 7: Event loop management
- Week 8: Events and final state machines
- Week 9: Mobile robot control
- Week 10: Networking for embedded applications
- Final exams week: Projects due

Learning Outcomes

By the end of this course, you will learn:

- Design an embedded application for small and moderately sized problems
- Implement a given design in the C/C++ programming language
- Generate appropriate documentation for developed solutions
- Design and implement tests for a given component
- Explore existing documentation to describe and use existing libraries and frameworks

Canvas Demo