

EE P 520: SOFTWARE ENGIEERING FOR EMBEDDED APPLICATIONS

SPRING 2021 COURSE OVERVIEW

Tamara Bonaci tbonaci@uw.edu

EE P 520 – Software Engineering for Embedded Applications

- Remote learning course
 - Synchronous lecture on Wednesdays from 6:00-9:00pm using UW Zoom
 - Optional post-lecture Q&A on Wednesdays from 9:00-9:30pm using UW Zoom
- Instructor: Tamara Bonaci (<u>tbonaci@uw.edu</u>)
 - Office hours: Saturdays from 9:00-10:00am or by appointment
- Teaching assistant: Trung Le (tle45@uw.edu)
- Course material: Canvas and GitHub
- Course discussion board: Piazza
- Assignment grades: Canvas
- Course assignment submission: Canvas and GitHub

What is EE P 520, Software Engineering for Embedded Applications?

- Modern embedded systems programming starts with the hardware, and extends all the way to the cloud
- The resulting myriad of programming languages, libraries, tools, data structures, and algorithms may seem difficult for any one programmer to master
- In this course, we introduce the fundamentals of programming languages and software engineering common to all levels of embedded systems programming, giving students conceptual tools they need to tackle various projects.

More specifically, this course:

- Takes a detailed look at two programming languages, C and C++;
- Teaches students how to use build tools, version control, and advanced editors;
- Describes how to use and create software libraries; and ties everything together with a more substantial software engineering project

EE P 520, Spring 2021 – Course outcomes

- At the end of this course, you should be able to:
 - Design an embedded application to 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

(Expected) Course Progression

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

Course Logistics

• Course will be graded based upon:

- Homework: 60%
- Project: 35%
- Participation: 5%

Course Logistics - Homework

- Seven programming assignments in this course
- The first six assignments are required
- Last assignment is optional, and can be used to replace your lowest score on HW1-HW6
- Homework assignments will be assigned every Thursday, and they will be due on Sunday the following week
- Please turn your submissions via GitHub. We clone your code and run unit tests in a Docker container

Overview of homework topics:

•HW1: Setup and getting started with GitHub, C/C++ and testing.

•HW2: C programming

•HW3: ADT

•HW4 : C++ programming

•HW5: Standard Template Library (STL) - 2 weeks

•HW6: Event Loop Management - 3 weeks

•Optional HW7: TBD - 3 weeks

Course Logistics - Project

- Individual project
- Assigned in week 7, and due in the finals week

Course Resources

- The Elma process manager
- Prof. Klavins' <u>Dockerhub</u> account
- We will be using <u>Github</u> for version controlling all of our code
- Please use the Visual Studio Code to edit your code for this course
- <u>Google Test</u> unit testing environment
- <u>Docker</u> container creation and management system
- <u>C++</u>language documentation
- Doxygen API documentation generator

Course References

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

Course Material

- If you can, try to "attend" lectures because:
 - Lectures will likely cover more than provided in lecture notes, and the provided references
 - Lectures will focus on "big-picture" principles and ideas
 - Your colleagues will likely start interesting discussions during lectures
 - In-class activities and discussions they will start

LATE TURN IN POLICY



All assignments are due by 11:59pm on the assigned date (Sunday)

U	-0-

Late assignments will (generally) be dropped 5% per calendar day, and no submissions will be accepted after 14 days



If you have a meaningful reason for delay (e.g., illness) - come and talk to us



Exception to the late turn in policy: final projects must be turned in on time

Your Questions

