

# ECE 103: Engineering Programming

## Catalog Description

Introduction to software design, algorithms, data structures, and programming using the "C" language. Interfacing to sensors, actuators, and other hardware. Writing documentation and presenting technical content. Weekly lab.

Expected preparation: [ECE 102](#).

Credit hours: 4

## Goals

Students will have the ability to apply software design principles and basic programming techniques to solve a variety of engineering, scientific, and mathematical problems.

## Course Coordinator and Committee

Branimir Pejčinović (coordinator)

Melinda Holtzman

Phillip Wong

## Textbooks

Required:

*ECE 103: Engineering Programming*, zyBook interactive textbook, zyBooks (John Wiley & Sons)

Note: This is an instructor-customized version of their *Programming in C* zyBook.

Optional:

*Engineering Problem Solving with C*, 4th Edition, D. Etter, Pearson Prentice Hall, 2012.

The course instructor may choose to use a different textbook. Please check with your instructor before purchasing.

## Prerequisites

MTH 112 with a grade of C or better, or passing at the necessary level on the mathematics placement test (see PSU Math Department webpage at [pdx.edu/math](http://pdx.edu/math) for information).

## Course Learning Outcomes

At the end of this course, students will have the ability to:

1. Analyze specifications and develop algorithms
2. Write C programs to solve intermediate-level engineering problems
3. Document the software design process for a technical report
4. Use a toolchain to develop programs
5. Be able to troubleshoot and debug programs
6. Work on a team to design and write control code for hardware interfacing projects

## Topical Outline

- Software development process
- Algorithms and pseudocode
- Editing, compiling, running, and debugging programs
- Preprocessor, program layout, library functions
- Data types, variables, storage classes
- Console and file I/O
- Arithmetic, relational, and logical expressions
- Branching
- Loops and jumps
- Function design
- Arrays and strings
- Pointers, dynamic memory allocation
- Enumerations
- Structures
- Overview of sorting and recursion
- Overview of stacks, queues, and trees
- Overview of multi-threaded and parallel programming
- Hardware interfacing using C programs and a microcontroller

## Course Structure and Grading Criteria

Structure:

- Two 110-minute weekly lectures
- One 120-minute weekly lab (7 weeks)

Grading Criteria:

- 
- Textbook readings & exercises (zyBook)
- Homework assignments
- Lab participation and reports
- Project (includes demo)
- Exams

Grading criteria may vary with instructor. Refer to the individual instructor's syllabus for information on the grading breakdown (i.e., percentage weight for each category) and the grading scheme.

## Relevant Student Outcomes

The following student program outcomes are supported by this course:

(1) An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

(5) An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives

Prepared By: Phillip Wong

Last revised: 4/7/2023