Programming for Engineers -- Elective

Course Description

This course introduces students to the software development process in a variety of settings, so that they acquire a wide set of programming skills. The first half of the course uses the P5.js library and the JavaScript language, while the second half uses Python. Emphasis on robust program design and sound software engineering practices will be put throughout the course. Each instructional unit will take 3-4 weeks.

Course Objectives

- Write programs using variables, functions, recursion, conditionals, loops, lists, arrays and objects.
- Organize code hierarchically and make use of modules.
- Design, implement and debug programs of several hundred lines of code, spanning several files.
- Demonstrate familiarity with the functions in the standard library of one or more languages.
- Explain the basic structure and organization of a medium-size program.
- Make reasonable choices for algorithms and data structures when designing a program.
- Properly use documentation and online resources to resolve programming questions.
- Work efficiently in a team to build programs collaboratively.

Assessing Performance

Formative assessment includes worksheets and several practice activities for each lesson, and unit quizzes. Summative assessment includes a programming project at the end of each unit.

Course Essentials

Equipment	**Cost/Unit**
Reusable material for physical computing	$1,500
Classroom set of computers	$0 if you already have some, $500-600 per computer if you need to purchase
Software	All the software used in this course is open source and widely available free of charge, but installation of software in local computers is required.

First Semester

Unit 1: Programming Basics	Programming languages. Development tools. Program design. Intro to P5.js
Unit 2: Graphics Programming	Drawing shapes and images. Keyboard and mouse control. Image processing. Sprite movement.
Unit 3: Physical computing	Arduino Platform. Controllable LED circuits, Breadboarding and prototyping. Timing control. Inputs and data acquisition
Unit 4: Integration project	Integration of graphics and physical computing in a single project.

Second Semester

Unit 5: Python	Syntax. Conditionals. Loops. Functions. Objects. Libraries and packages.
Unit 6: Console-based programming	Console and file IO. Input validation. Floating point calculations. Numerical algorithms. String processing.
Unit 7: Data structures	Lists. Tuples. Dictionaries. Heaps, stacks and queues. Trees. Graphs.
Unit 8: Object-oriented design	Encapsulation. Polymorphism. Inheritance. Top-down design. GUI programming.
Unit 9: Algorithm design and analysis	Linear and binary search. Naïve sort. Quick sort. Merge sort. Recursion.