

Course Syllabus

INSTRUCTOR:

William P. Osborne
16606 226 Ave NE
Woodinville, WA 98077
phone: (206) 954-1815
email: posborne@lwsd.org

TITLE: Introduction to High School Java and C for Robotics and Engineering

DATES: Not yet determined

NUMBER OF CREDITS: 1 credit based on 10 hours of contact time with an optional 5 hours of lab.

INSTRUCTOR INFORMATION: BSEE and MIT from Seattle University, MBA and MS from Stanford University. Extensive industry experience. Currently computer science and engineering teacher at Redmond High School.

COURSE DESCRIPTION: This course is for teachers with little or no programming background who wish to teach computer programming, robotics, and engineering to hands-on learners. Teachers will learn the basics of programming in the Java language as it is used in college-level introductory computer science, and the similar C programming language as it is used for robotics and engineering.

COURSE OBJECTIVES: By the end of this course teachers will

1. Know how to install and use Processing, a programming system developed at the MIT Media Lab.
2. Be able to write working Java programs that can receive information from files, a keyboard, and a mouse and then process that information to produce rich text, images, sounds, and animations.
3. Understand that the C programming language used in robotics and electronics is similar to Java.
4. Know how to install and use Arduino, the programming system based on Processing designed specifically to create programs for robotics and other systems driven by an embedded computer.
5. Be able to write working C programs for the Arduino for the purposes of receiving and acting on information for the control of motors, lights, sounds, and many other electronic and mechanical devices.

MATERIAL COVERED:

The subset of the Java and C languages to be covered are:

1. The writing, compiling, running, and debugging of programs.

2. The programming statement - a command to carry out some specific action such as show an image or perform a mathematical operation. The programming statements to be learned are for:
 - a. the creation of variables and the assignment of values to those variables
 - b. the evaluation of mathematical and logical expressions
 - c. the control of program flow based on the evaluation of logical expressions
 - d. the repetition of sets of programming statements until some condition has been met
 - e. the input of information from keyboards, mice, switches, and sensors
 - f. output for the presentation of information, communication, and the control of motors and other electronic component
3. The grouping of programming statements into methods to serve specific purposes.
 - a. Required methods – each program must have the following two methods:
 - i. `setup()` – contains statements to perform all tasks necessary to allow the program to run. Examples are to create a window on the computer screen suitable for showing an animation, and to initialize the control of a servo motor for a mechanical arm
 - ii. `loop()` – contains statements executed over and over. Typically the `loop()` method looks for information from the user or from a sensor, evaluates this information, then takes action
 - b. Helper methods – added as needed to perform specific tasks. If a program needs to be able to decode a message from a remote control, for example, the programming statements to do this would likely be placed in a helper method

In addition to the languages teachers will learn

1. How to connect an Arduino to a computer and upload a program.
2. Which parts of the C language are specific to the Arduino and how to use them.
3. How to test an Arduino, including how to run the example programs that come with the Arduino development environment.

SESSION CONTENT, ASSIGNMENTS, and ASSESSMENTS:

Day	Content	Assignments and Assessments
1	Installation of Processing software Program creation, editing, saving, and running. <code>setup()</code> and <code>loop()</code> methods, and <code>String</code> variables	Assignment: Several small programming projects. Programs to be submitted electronically for grading.
2	integer and boolean variables, mathematical operations, <code>for</code> loops	Assessment: Short quiz on <code>setup()</code> , <code>loop()</code> , and <code>String</code> variables.

		Assignment: Self-assessment problems plus writing a program that uses <code>for</code> loops to perform calculations and draw geometric figures.
3	branching, logical operators, helper methods, <code>while</code> loops	<p>Assessment: Short quiz on <code>for</code> loops and math operators.</p> <p>Assignment: Self-assessment problems plus writing a program with a user menu. Items on the menu perform different tasks. The programming statements in each task are in separate helper methods.</p>
4	<p>Installation of Arduino software.</p> <p>Similarities to Processing, Arduino-specific methods, sensing and controlling electronic devices</p>	<p>Assessment: Short quiz on <code>while</code> loops and helper methods.</p> <p>In-class assignment: Write a program in Arduino C that causes a motor to turn and a light to blink in response to a button being pushed.</p>

TRANSFERRING SKILLS:

1. The class structure models the hands-on learning environment of successful high school programming classes.
2. Teachers will learn to critique, correct, and debug student programs by performing that task for each other.
3. Teachers will work with actual programs written by high school students.