Introduction to Computer Science

Course Syllabus

Roosevelt HS

2011-12

Introduction to Computer Science

1 semester | ½ credit (Occupational Education)

Andrew Davidson
ahdavidson@seattleschools.org
Room 319 | 206.252.4853

DESCRIPTION

This course is an introduction to computer science and software engineering for all students interested in developing software applications, not just using them.

Through a project-oriented approach, students will explore a variety of programming systems and languages to create interactive applications and systems. By collaborating in a hands-on environment, students will learn problem solving, software design, debugging strategies, and the foundations of computer science (data structures, procedures, and algorithms). Students will work on projects (both individual and team) in the areas of graphics and games, animation and art, electronics systems, and interactive fashion, all using open-source software tools such as Scratch, Arduino, Processing, and Python.

PREREQUISITES

Basic familiarity with computers and software applications, plus a curious spirit and a willingness to experiment and learn.

OBJECTIVES

Upon successful completion of this course, students should:

- Understand basic principles of thinking and solving problems with computers and computation.
- Be able to specify procedures for solving problems and the concepts and practice of designing and implementing algorithms.
- Recognize, analyze, and correctly use the basic elements of computer programs, such as statements, variables, events, conditionals, loops, timers, and arrays.
- Grasp and utilize the fundamental concepts of computer science including data types, control structures, operators, functions, and libraries.
- Have a working knowledge of computer system design and the practice of software engineering, software tools, debugging, and documentation.
- Be able to design, plan, prototype, implement, test, and document a reasonably complicated software (and hardware) project.

TOPICS

A summary of the units of study and topics for the course is listed below. Detailed information about assignments and assessments may be found on the course website, as noted below.

Unit	Topics	Duration
Computational Thinking	Problem Solving System Design Algorithms	1 week
Elements of Programming	Control Flow Iteration Variables Events Conditionals Timers Arrays	4 weeks
Computer Science Fundamentals	Data Types Control Structures Operators Functions Libraries	3 weeks
Software Engineering	Software Design Programming Tools Debugging Documentation	4 weeks
Final Project	Concept & Proposal Design Documentation Project Plan Prototype Implementation User Testing Communication	6 weeks

MATERIALS

There is no textbook for this course, although there will be reference books available in the classroom. Most of the **teaching materials** (handouts, presentation slides, resource lists, assignments, etc.) will be found online, either on a website that I have developed specifically for this course (see below), or other public website resources.

Please be sure to have a **composition book** for the class (as well as writing instruments, of course), as you will need it for notes and keeping a journal. If you intend to work on your projects outside of school, you will need a flash drive to transport digital files back and forth.

We will be using many different **software tools** in the course. All of them are installed on the computers in the lab. Most of the software is "open source," which means it is free and available for anyone to download and use on their own computers. So you can also work at home or anywhere else you have access to another computer.

COURSE WEBSITE

You can find complete information about this course online, at the website I have developed for all of my classes. **http://rooseveltcs.org/courses/intro-to-cs/** is the link for this course, but you will find useful information on other parts of the website as well.

I intend to use the website, with its announcements blog, as the main source of all assignments, reference materials, and communications. Grades will be posted on the Source, but I won't be posting detailed information about assignments there.

The website is publicly accessible, so you can view it from home or any internet-connected computer.

ASSIGNMENTS

This course is based primarily on the philosophy of **project-based learning**. This means that students will learn by designing and implementing programming projects.

There will be regular **assignments** to complete, typically one per week or every other week, depending on the topic. These assignments will involve applying the principles and techniques of computation, programming, and computer science in increasingly complicated exercises and small projects.

The **final project** is an investigation of how to design and implement a software application and will also afford the opportunity to experiment with basic digital electronics. This project will require students to design, plan and implement a software project of their own choosing. Students will be encouraged to work in small teams on this project, to gain experience in working collaboratively, much as software projects are developed in real companies.

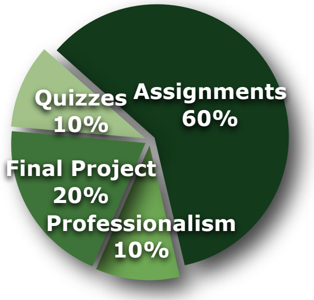
There will also be a number of short **quizzes** during the semester to reinforce understanding of the concepts and technical details of programming.

Finally, a component of all Career and Technical Education (CTE) courses is a demonstration of **professionalism**. Students are expected to actively participate in class, show leadership, and exhibit responsible behavior in preparation for life beyond high school. There will be activities and tasks in the classroom that emphasize these skills.

GRADING

Grades will be based on the components as shown below.

• **Assignments**	60%
• **Final Project**	20%
• **Quizzes**	10%
• **Professionalism**	10%



Grades will be posted on the Source, generally within two weeks of the due date of the assignment.

LATE WORK

This course is structured so that you can accomplish all of the required work during class sessions, because much of it requires software and supplies that are in our lab and not generally available on other school computers.

But things always come up, from absences to illness to just needing extra time. Late work will be accepted, without regard to cause, because I would rather have you do the work and learn the material we cover than not.

However, to be fair and to encourage you to keep up with the work, late assignments will be **marked down 10%**. You can submit late work **not more than two weeks** past the due date, without extenuating circumstances.

POLICIES

A complete list of general course policies can be found in the attached "**Course Policies**" document.

SUPPORT

I will be available most days during lunch periods and after school for extra lab time or individual help. Email (*ahdavidson@seattleschools.org)* is also a great way to get in touch with me for questions or problems.