AP® Computer Science A Syllabus

Course Overview

The focus of this course is to introduce students to one of the fastest growing job markets in the United States. It will increase students' problem solving skills, abstraction abilities, and critical thinking. The content and objectives of the course are aligned with the CollegeBoard *AP Computer Science A Course Description.* The course focuses on an object-oriented approach to problem solving using Java. It includes the study of common algorithms and the use of some of Java's built-in classes and interfaces for basic data structures. All students are required to take the AP Exam at the end of the course.

Course Philosophy

This course is designed to teach computer programming skills to both new and experienced programmers. Programming is not a spectator sport; students must write and debug programs to *fully understand* programming. Our school is a one to one school and each student will be issued a laptop for use in and out of class. Students will work individually and in groups to learn and improve their knowledge of Java and programming in general.

The majority of the class, over 90%, will be hands on in class programming exercises and labs. As a mathematics teacher, the majority of my lab assignments are to implement various mathematical algorithms. A sample of, but not all, programming activities are included in the course outline.

Assessments

Assessments will be a combination of graded labs and tests. All tests resemble the AP examination.

Course Materials

Primary Text: Farrell. *Java Programming, 8th Edition.* Boston, Cengage, 2016.

Laptop Computer: Each student will be issued a laptop computer with Java and jGrasp installed.

Portfolio: Students are required to maintain an organized portfolio of their work. The portfolio will be both paper (class notes and handouts) and digital (labs and projects).

Auxiliary Materials: Released exams, study guides, old textbooks and other materials available on the internet will be used throughout the year to prepare students for free response and multiple choice style questions.

Course Outline

Times are approximate and vary based on student mastery – remaining time is used to review and practice for the AP® exam.

Creating Java Programs (2 weeks)

An introduction to programming. Students will learn basic programming terminology and apply this to the Java programming environment. They will also learn how to create a simple Java application that produces output to the console. Students will learn the basic components of a Java application and how to compile and run a Java program. Programming activities: Hello World, learning the difference between `println` and `print.` Debugging provided code.

Using Data (3 weeks)

Introduction to the eight primitive data types in the Java language. Students will learn to work with integer, floating-point, Boolean, and character values. Arithmetic and comparison operators as well as type conversion are introduced. Finally, students will learn to create input using the `Scanner` class. Exception handling will be covered at an introductory level only. Programming activities: Basic arithmetic operations, generating random numbers between specified values, solving for the roots of a quadratic equation.

Using Methods, Classes and Objects (3 weeks)

Introduction to the creation of classes, variables, and methods. Students will learn to create methods that accept arguments and return values. They will learn to create a class composed of instance variables and methods. Students will create and invoke constructor methods to initialize instances of a class. Students will be introduced to blocks and scope, overriding, and overloading. Students will learn to create overloaded methods and constructors. They will learn about `static` variables and how to create constants using the `final` keyword. Finally, students will learn to use prewritten classes in the `java.lang` and `java.util` packages, and other packages.

Making Decisions (3 weeks)

Introduction to decision structures using the `if`, `if…else`, and `switch` statements. Students will learn to execute program statements based on the result of a Boolean expression. They will also learn to use the logical operators AND, OR, and NOT, as well as the conditional operator. Students will learn to avoid common programming errors when making decisions. Programming activity: Convert raw test scores into projected AP scores and letter grades, Condo cost based on specifications.

Loops (3 week)

Students will learn to create definite and indefinite loops using the `while` statement. They will learn to use Java's accumulating and incrementing operators. They will use `for` loops to create a definite loop and `do…while` loops for use when a posttest loop is required. They will learn how to create nested loops and how to improve loop and algorithm efficiency. Programming activities: Converting the factorial of a number, compound interest, annuities, Newton's method for finding zeros of an equation.

Characters, Strings and the `StringBuilder` (3 weeks)

Introduction to character strings in Java. Students will learn to use the `Character`, `String`, and `StringBuilder` classes for single characters, immutable strings, and mutable representation of a character strings respectively. Programming activity: Writing business letters, writing in Pig Latin

Arrays (6 weeks)

Introduction to the concept of arrays. Students will learn to create arrays of primitive data types and objects. They will work with arrays by searching, sorting, and passing them to methods. Students will also learn how to use one dimensional and multidimensional arrays, and the `Arrays` and `ArrayList` utility classes. Programming activities: Dot product, matrix multiplication, bowling teams, sequential/linear search, binary search, insertion sort, bubble sort, selection sort, merge sort

Inheritance (4 weeks)

Expand on the concepts of inheritance, polymorphism, and information hiding, which were introduced at the beginning of the course. Students will learn to derive classes and to access constructors, data fields, and methods of a superclass. Students will learn to create abstract classes. Dynamic method binding enables programmers to declare an object using the name of a superclass and then instantiate a subclass. A superclass can also be used as a method argument or as an array type. Students will learn about the `Object` class and how to use the `toString()` and `equals()` methods. Finally, students will learn to create interfaces and packages. Programming activities: Dinner party, calculate shipping cost

AP Computer Science Labs

I have integrated the AP Computer Science A Labs into my course at appropriate times based on their content, which account for a minimum of 20 hours of hands-on lab work (e.g., four hours on Magpie labs, six hours on PictureLab labs, and ten hours on the Elevens labs). Students complete the Magpie labs to help them develop their conditional statement skills. They complete the PictureLab labs to practice two-dimensional array algorithms. The Elevens labs is distributed across the school year. Students complete all of the required activities of the AP Computer Science A Labs.

Ethical and Social Implications of Computer Use

Responsible use of computer systems is integrated throughout the course. Students are encouraged to bring in and discuss current articles highlighting ethical and social implications of computer use at any time. I also highlight current events and policy discussions as they occur.