AP Computer Science A

1.0 Credit, 1 period per day for 180 days

- Mac or PC required in addition to iPad
- 20 hours of individual student programming time ("Labs") required

Resources:

- greenfoot.org - free software to practice Java programming with visual tools
- drjava - free interactive development environment for console-based Java programming
- codingbat.com - free site to practice Java logic and programming
- Karel J. Robot - free site to program a virtual robot using Java
- Decoding AP Computer Science A - free ebook by Moksh Jawa with corresponding website

Units

1) Introduction to Java features (25 days)
2) Control structures (25 days)
3) Classes and objects (25 days)
4) Arrays and arraylists (25 days)
5) Inheritance and polymorphism (15 days)
6) Recursion (15 days)
7) Sorting and Searching (25 days)
8) Social implications of programming (10 days)

Totals to 165 days to complete required materials before the AP exam which is typically given in the beginning of May.

Time after the exam will involve students doing additional programming in "RoboWar" where teams program a virtual robot to compete against others along with other programming projects.

Unit 1 Introduction to Java features

Time: 25 class periods

Know:
object-oriented programming, packages and classes, comments, keywords, identifiers, types, hexadecimal, octal, decimal, variables, operators (arithmetic, relational, assignment, increment/decrement, precedence), input, output, escape sequences, IDE (interactive development environment)
Understand:
What are the features of a modern high-level computer programming language?
Do:
How do low-level and high level programming languages compare to each other?
What are the advantages of object-oriented programming?
How do packages and classes make programming more efficient?
How is an Interactive Development Environment set up and used to write a Java program?
What are common keywords used in Java?
How is information stored in a computer?
How are operations performed on data?
How is information entered and displayed in a computer program?
Computer Lab Projects: Create a program to solve kinematics physics problems, Magpie Lab (activities 1-5)

Unit 2: Control Structures

Time: 25 class periods

Know: If/Elseif statements, While loop, For loop, iteration,
Understand: How are control structures implemented to make repetitive and conditional tasks execute?
Do: How are If statements used? When should a While loop be used? How is iteration accomplished using For Loops? How can a complex problem be solved using control structures? Computer Lab Project: Program a robot to interact with its environment using if, while and for structures. Strings and randomness – HexChecker, MorseCode, Rock-Paper-Scissors, and Etch-A-Sketch

Unit 3 Classes, Objects, and Methods

Time: 25 days

Know:
class, object, method, public, private, static, encapsulation, scope, constructor, accessor, mutator, method overloading
Understand:
How can a programmer create methods beyond the simple main method?
Do:

How are classes, objects, and methods related to each other and how do they fit into a Java program?
How are variables restricted to different parts of a program?
How can an overloaded method be created?
How do public, private, and static alter the scope of a variable?
How is a complex program created with multiple methods?

Computer Lab Project: Create a simulation of motion and collisions.

Unit 4: Arrays and arraylists

Time: 25 days

Know:
one-dimensional array, multidimensional arrays, index, array list, List<E> interface, iterator, for-each traversal,
Understand: How are arrays used to store and analyze large amounts of data?
Do: What is an array? How is an array created and read? What does it mean to traverse an array? How is a for-each loop used to analyze an array? How can multivariable linear equations be solved using arrays? Computer Lab Project: Analyze acceleration data from a punching bag to determine how many times it was hit.

Unit 5: Inheritance and polymorphism

Time: 15 days

Know:
Superclass, subclass, inheritance hierarchy, polymorphism, type compatibility, abstract classes, interfaces
Understand: When are subclasses used and how do they relate to superclasses?
Do: How does extending a class add functionality to a class? How does inheritance affect what variables are available to a class? How is polymorphism similar and different to an overloaded method? How do variables get cast when using super and sub classes? When would an abstract class be used? How is an interface implemented? Computer Lab Project: AP CSA Elevens Lab

Unit 6: Recursion

Time: 15 days

Know: recursion and related methods and algorithms
Understand: How can recursive methods be used to solve problems?
Do: How are recursive methods similar and different to iterative methods? How is the Merge Sort used? Computer Lab Project: Create algorithms for determining physical and mathematical constants.

Unit 7: Searching and Sorting

Time: 25 days

Know: recursion and related methods and algorithms
Understand: How can recursive methods be used to solve problems?
Do: How are recursive methods similar and different to iterative methods? How is the Merge Sort used? Computer Lab Project: PictureLab - students manipulate data in a picture.

Unit 8: Social implications of computing

Time: 10 days

Know:
copyright, "do no harm" concept, net neutrality, privacy, social media
Understand: What are the ethical and social implications involved with the use and programming of computers?
Do: What is open-source coding?   How is privacy protected (or not) on the internet?   How can technology be used ethically?   What are the dangers and benefits inherent in social media?   Computer Lab Project: Examine resources at http://jodypaul.com/sweng.html