

# AP Computer Science A

## 1. Course Overview

Welcome to AP Computer Science A. I hope this class will be interesting and enjoyable as I will do my best to make it so. In this class, you will learn how the parts of a computer work together, what it means to program a computer using the Object-Oriented paradigm and the Java programming language, and give you ample opportunity to solve problems using a computer.

The AP exam for this class takes place in early May of each year. The specific date, time and place will be given to you at a later date. This course roughly equivalent to a first semester college computer science class (exact curriculum depends on specific college) and is structured to both prepare you for passing the AP exam as well as be successful in your future computer science courses.

## 2. Course Plan

The AP Computer Science course taught at our school is considered a “1<sup>st</sup> course” as would be a college-level class because most of the students have not had any experience with programming and computer science. Below is the sequence of our AP Computer Science A course. The exact number of days may vary according the students in the class. Some classes progress much quicker than others and would be able to do more enrichment programming activities than given below.

### Matrix of Curricular Requirements and Codes Used Below:

C2. The course includes all of the topics listed in the “Computer Science A” column of the Topic Outline in the <i>AP Computer Science Course Description</i> .	C3. The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.
C4. The course teaches students to use and implement commonly used algorithms and data structures.	C5. The course teaches students to develop and select appropriate algorithms and data structures to solve problems.
C6. The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. The course teaches students to use the standard Java library classes from the AP Java subset delineated in Appendix A and B of the AP Computer Science Course Description.	C7. The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.
C8. The course teaches students to read and understand a large program consisting of several classes and interacting objects, and enables students to read and understand the current AP Computer Science Case Study posted on AP Central.	C9. The course teaches students to recognize the ethical and social implications of computer use.

## Unit 1 – An Introduction to Hardware, Software and the Internet ( 4 days: 4 (class) and 0 (lab) )

### Topics

Hardware Overview

Software Overview

What Do Programmers Do?

Data Representation

Computing Systems

Responsibilities of a Computer Science Student and Computer Scientist

Labs: None

Topic Outline (C2)	Other APCS Curricular Requirements
Section III: H1, H2 Section VI: A1-3, B1-3, C1-2, D1-4	C7, C9

Commentary: This serves as an introduction to Hardware, Software, and Systems. These topics will continue as a discussion topic as the year progresses and as the APCS student gets a better understanding of how all of this fits together. I have roles and responsibilities as a computer user for my lab and this discussion leads itself to the roles and responsibilities of a computer user/Computer Scientist in the world.

## Unit 2 – An Introduction to Software Development ( 6 days: 2 (class) and 4 (lab) )

### Topics

Compilers and Interpreters

Software Components and Packages

Object-Oriented Programming

Labs: Three ways to say hello (from text)

Introduction to Turtle Class (from Java au Naturel)

Topic Outline (C2)	Other APCS Curricular Requirements
Section I: A5 Section II: A1, B3 Section III: C Section VI: B1-3	C6 (intro), C7, C8 (intro)

Commentary: This unit gets the students hands on the computer, typing in small programs given to the student, working with the IDE (JCreator) and gives some “Big Picture” views of where we are going this year.

## Unit 3 – Objects and Classes ( 9 days: 3 (class) and 6 (lab) )

### Topics

Objects in a Program

Classes

Fields, Constructors, and Methods

Inheritance

Labs: First Steps (from text)

Drawing an Octagon with Turtle Object (from Java au Naturel)

GridWorld – Observing and Experimenting

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1 Section II: A1, B1, B3, B5, C Section III: A1, A3, C	C6 (intro), C8 (intro)

Commentary: The labs of this unit have the student do more than copy from the book, filling in missing code in the “First Steps” lab and writing a complete program (main) using the Turtle object from Java au Naturel which is a drawing device with a formal list of methods. This is also the first place where students will work with the GridWorld Case Study.

## Unit 4 – Algorithms (3 days: 3 (class) and 0 (lab) )

### Topics

Properties of Algorithms

Iterations

Recursion

Working with Lists

Labs: None

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1, B3 Section II: B4 Section IV: C Section V: B1, B2	C4 (intro)

Commentary: An introduction to “thinking and problem solving”...something the high school students often do not have the patience or training for when entering this course. This unit is “out of the lab” so the students are forced to think without a computer in front of them.

## Unit 5 – Java Syntax and Style (4 days: 3 (class) and 1 (lab) )

### Topics

Using Comments
Reserved Words in Java
Syntax vs. Style
Statements, Blocks, and Indentation
Labs: Correcting Syntax Errors

Topic Outline (C2)	Other APCS A Curricular Requirements
None	None

Commentary: Although not formally in the Topic Outline, the use of comments and consistent style, including indentation, is an essential skill for any Computer Scientist.

Note: 2 days are reserved for the first exam within these days.

## Unit 6 – Data Types, Variables, and Arithmetic ( 8 days: 4 (class) and 4 (lab) )

### Topics

Fields and Local Variables
Primitive Data Types
Strings
CONSTANTS
Scope of Variables
Arithmetic Expressions
Compound Assignment and Increment Operators
Converting Numbers and Objects into Strings
Labs: Pie Chart (from text) Triangle Calculator

Topic Outline (C2)	Other APCS A Curricular Requirements
Section I: A1, A5 Section II: A1, B1-3, C (intro) Section III: A2, B1-3, C, H2 Section IV: A, B	C3, C4, C6

Commentary: In this unit, the student finally needs to be cognizant of the data being represented, a huge step in the process of learning concepts of Computer Science. The classes that are given are in a template form with pre- and post-conditions provided for the problem description.

## Unit 7 – Boolean Expressions and `if-else` Statements ( 10 days: 4 (class) and 6 (lab) )

### Topics

`if-else` statements

boolean Data Types

Relational and Logic Operators

Order of Operators

Short-Circuit Evaluation

`if-else-if` and nested `if-else` statements

The `switch` Statement

Labs: Moving CD's with the Vic Controller (from Java au Naturel)  
Rolling Dice (from text)

JavaBat: Begin assigning methods from [www.javabat.com](http://www.javabat.com) Three methods each week out of the Logic category.

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1 Section II: A1, B1-4 Section III: A1-3, B1-3, C, F1-2 Section IV: A, B	C3, C6

Commentary: The student can now do more interesting work because decisions can be made.

Logic becomes very important and this is often the first formal exposure to logic for many of the students in class. I have found that the Vic (basically a controller of a sort) to be an effective way of teaching logic, visual so that the students can see results while at the same time rich with a collection of methods that are quite powerful.

In this unit I also start the students practicing more out of class with the JavaBat website. This is a website that provides programming practice so that the student can become much more fluent in coding in Java. The methods start small and simple in the logic category, but become increasingly more difficult as the year goes by in the categories of String, Arrays, Recursion, and AP type questions.

## Unit 8 – Iterative Statements: `while`, `for`, `do-while` ( 12 days: 6 (class) and 6 (lab) )

### Topics

The `while` and `for` Loops

The `do-while` Loop

Using `return` and `break` in loops

Nested Loops

Labs: Moving CD's iteratively using the Vic Controller (from Java au Naturel)  
 Perfect Numbers (from text)  
 Sierpinski Triangle

JavaBat: Three methods each week from the Logic category.

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1 Section II: A1, B1-4 Section III: A1-3, B1-3, C, F1-2 Section IV: A, B	C3, C4, C5, C6

Commentary: In the Unit 7 labs involving the Vic Controller, students must copy and paste in order to do something iteratively and become “hungry” for a way to repeat something. A perfect placement in the sequence of topics, much of what the students worked on in Unit 7 becomes reusable, with a simple change from an `if` to a `while`. The `for` loop is introduced here but not used extensively until needed when working with `Strings` and `Arrays/ArrayLists`.

Note: Two days within unit 8 are used for the second exam.

<h2 style="margin: 0;">Unit 9 – Implementing Classes and Using Objects</h2> <h3 style="margin: 0;">( 13 days: 9 (class) and 4 (lab) )</h3>
--

<b>Topics</b>
Public and Private Features of a Class
Constructors
References to Objects
Defining Methods
Calling Methods and Accessing Fields
Passing Parameters to Constructors and Methods
return Statement
Overloaded methods
Static Fields and Methods
Labs: Snack Bar (from text) Complex Number Calculator
JavaBat: Three methods each week from the Logic category.

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1-3, 5, B1-4 Section II: A1, B1-4, C Section III: A1-3, B1-3, C, E, F1-2 Section IV: A, B	C3, C4, C5, C6

Commentary: My experience with students when first introduced to the idea of designing classes, and they are in front of a computer, is that there is not much thought put into the process of design. One entire week of this unit will be spent designing and writing classes away from the computer screen so that the students can get some kind of process down: Constructor(s), accessor methods, modifier methods, data fields. In my experience, if this is not solidified early, the coming units just get more and more difficult.

## Unit 10 – Strings

( 11 days: 7 (class) and 4 (lab))

### Topics

Literal Strings

String Constructors and Immutability

String Methods

Formatting Numbers into Strings

Extracting Numbers from Strings

Character methods

Labs: Exploring the String class' Methods  
Lipograms (from text)

JavaBat: Three methods each week from the String1 category.

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1-3, 5 Section II: A1, B1-4, C Section III: A1-3, B1-3, C, E, F1-2 Section IV: A, B	C3, C4, C6

Commentary: It is essential to be able to use methods from a Standard Java Class. This is the first real in-depth exploration of a class from the AP subset and the first lab explores the methods in the subset, but also explores some of the methods that are useful but not in the subset.

Note: 3 days are used for review for the final exam. This unit ends the study for the first semester.



## Unit 11 – Class Hierarchies and Interfaces ( 14 days: 4 (class) and 10 (lab) )

### Topics

Class Hierarchies and Polymorphism

Abstract Classes

Invoking Superclass's Constructors

Calling Superclass's Methods

Interfaces

Labs: Dance Studio (from text)  
GridWorld – Extending the Bug class

JavaBat: Three methods each week from the String2 category.

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I: A1-3, 5, B1-4 Section II: A1, B1-4, C Section III: A1-3, B1-3, C, E, F1-2 Section IV: A, B	C3, C4, C6, C8

Commentary: This is the point when the students will final have enough background to tackle some of the interesting problems with the GridWorld Case Study. With an introduction early on in the course, the case study will be brought back with an in-depth look at enhancing the program with classes that inherit from the Bug class.

## Unit 12 – Arrays and ArrayLists ( 18 days: 6 (class) and 12 (lab) )

### Topics

One-Dimensional Arrays

The `ArrayList` class

The `ArrayList` Constructors and Methods

Comparison: Arrays vs. ArrayLists

`for` loop and `for-each` loop: Compare and Contrast

Inserting and Removing Objects

Labs: Creating an Index for a Document (from text)

GridWorld – The GridWorld Environment Classes and Interfaces

GridWorld – Interacting Objects

JavaBat: One method each week from the Array1 category/Two from the Array2 category

Topic Outline (C2)	Other APCSA Curricular Requirements
Section I : A – B Section II: A1, B1-4, C Section III: A – G Section IV: A- C	C3, C4, C5, C6, C8

Commentary: In this unit, the student at our school will fully immerse herself in the GridWorld Case Study. It is essential for students to see and work with a program of this magnitude because as a software engineer in my past career, I saw the large-scale programs and interactions between modules in real-life situations (designed engine control software and telemetry radio software for the Chevy Indy 500 race car program).

This is also where the student gets a heavy duty exposure to using Arrays and ArrayLists and deciding which data structure better suits the needs of the application.

Note: Two days are for an exam.

## Unit 13 – Searching and Sorting ( 15 days: 6 (class) and 9 (lab) )

### Topics

`equals`, `compareTo`, and `compare`

Sequential and Binary Search

Quadratic Sorts: Selection Sort, Insertion Sort

Recursive Methods

Divide-and-Conquer Sorts: MergeSort and QuickSort

Labs: Keeping Things in Order (from text)

Benchmarks (from text)

Simple Database – Your Addressbook

JavaBat: Two from the Array2 category each week/ One from Recur1

Topic Outline (C2)	Other APCS A Curricular Requirements
Section I : A – B Section II: A1, B1-4, C Section III: A – G Section IV: A- C Section V: A - C	C3, C4, C5, C6, C8

Commentary: This is the unit where it all comes together for the student (hopefully!!). Use of Inheritance, Arrays, ArrayList, String, and all of the control structures to keep a sorted database of objects (contacts). It is here where we compare efficiency of algorithms, both searching and sorting, by looking at how long each algorithm takes given a particular data set and which algorithms are best under possible worst-case scenarios

<b>Unit 14 – Streams and Files</b> <b>( 5 days: 2 (class) and 3 (lab) )</b>
--

<b>Topics</b>
---------------

Pathnames and the <code>java.io.File</code> class
---

Reading from a Text File
--------------------------

Writing to a Text File
------------------------

Labs: Hangman
---------------

Topic Outline (C2)	Other APCSA Curricular Requirements
None	C3, C4, C6, C8

Commentary: Although file I/O is not part of the AP Computer Science A topic outline, it is an important concept in real-world programming problems, whether it be a PC based program like a text editor or an instrument like a data logger put out into the wilderness to measure snow depth. The Hangman game will read from a text file that has 20,000 words and the player will have to guess the word.

**Post AP Exam – Topics not included in the AP Computer  
Science A outline but are Important.**  
(number of days varies depending on the school calendar)

**Topics**

Model-View-Controller

Multi-Dimensional Arrays

AP Computer Science Annual Project

Commentary: The AP Computer Science Annual Project is a large program that shows use of multiple classes and use of the data structures taught in the course. This is often a game of some sort. I also allow students to take on a project that uses Visual Basic or another programming language, giving the student a better perspective of computer science as a whole.

---

Conclusion: The AP Computer Science A course at our school starts with the understanding that the student has little or no experience with computer science. By the end of the year the students have had a course that includes an in-depth study of Object-Oriented Program design, Program Implementation, Program Analysis, the Standard Data Structures of the AP Computer Science A course, the Standard Algorithms of the AP Computer Science A course, and a look at how all of the pieces fit together, both electronically as a piece of equipment and socially as part of a world that is getting more and more computer software that controls our very existence.

This is accomplished through a variety of application areas from the Vic class that is somewhat like modeling a robot to a Database program that is similar to a “Phonebook” on a typical cell phone. Some of the programs are visual by nature (Vending Machine) and some are more abstract (Complex Number Calculator).

Students in our AP Computer Science course get experience programming “in the small” through coding practice on the JavaBat website and “in the large” with the enhancements to the GridWorld Case Study.

Overall, there is a nice balance of “class” time where there is lecture, modeling good programming practice, and written assessment...and “lab” time where students are allowed to explore the ideas on their own and find the “addictive” qualities of this field of study!!

### 3. Grading Policy

Our department has the belief that as a student progresses along the course offerings of the school, the emphasis on grading should become more and more weighted towards knowledge. The AP Computer Science grading policy is as follows:

---

1) 30% - Programs
<u>Includes:</u>
30% - Programs

---

2) 60% - Tests and Quizzes
<u>Includes:</u>
40% - Unit Tests
20% - Quizzes

---

3) 10% - Homework and Classwork
<u>Includes:</u>
10% - Homework and classwork

---

Grading scale	92 - 100	A	77 - 80	C+
	90 - 92	A-	70 - 77	C
	88 - 90	B+	67 - 70	C-
	82 - 88	B	60 - 67	D
	80 - 82	B-	<60	F

### 4. Materials required for this course:

- Textbook (provided) Litvin, Maria and Gary, Java Methods A & AB, Object-Oriented Programming and Data Structures, AP Edition. Skylit Publishing, 2006.
- Online Text: Java au Naturel, Jones, William C., Fourth Edition, 2006  
<http://www.cs.ccsu.edu/~jones/book.htm>
- Copy of GridWorld Case Study Narrative
- Copy of GridWorld Case Study Quick Reference Appendix A, B, C, E

### 5. Web-based stuff:

---

Web Site	From my home page, click on <u>Me the Teacher</u> and the other links should be self-explanatory. Here you should be able to find: <ul style="list-style-type: none"><li>➤ your unit assignments,</li><li>➤ necessary files for the labs,</li><li>➤ a link to the lessons (PowerPoint),</li><li>➤ your grades using your student ID.</li><li>➤ links to other Computer Science related websites that we will be using.</li></ul>
----------	--

---