

# AP<sup>®</sup> Computer Science "A" 2009-2010 Syllabus

<b>Teacher</b>	Wiley Sullivan. <a href="mailto:wsullivan@farmersvilleisd.net">wsullivan@farmersvilleisd.net</a> B.A. Baylor University, M.S. University of North Texas AP Computer Science Website at <a href="http://fisdapcomputerscience.blogspot.com">http://fisdapcomputerscience.blogspot.com</a>
<b>Prerequisites</b>	Algebra I and Geometry (or concurrent enrollment in Geometry with instructor permission) Students who are willing to accept the challenge of rigorous academic curriculum.
<b>Grades</b>	<p>In order to maintain fairness, course grades in Honors, Dual Credit and Advanced Placement courses receive additional weight and five points added to the grade. This is to acknowledge the fact that these courses are more difficult than regular courses.</p> <p>Grades for this class will entail written homework assignments, programming projects, in-class demonstrations/presentations, quizzes, and tests. All of these will have an assigned point value. The students current/final grade will be determined by the number of points earned divided by the total number of points possible. Students have chances to obtain bonus points through bonus assignments determined by the teacher.</p>
<b>Access to Computers</b>	AP Computer Science is taught to 6-12 students in a classroom containing 18 Pentium IV 2.8GHz computers with 2 GB RAM running Windows XP Pro. Students have the opportunity to program a minimum of three hours a week in class and/or in other computer labs during the course of the school day and after school.
<b>Texts &amp; Software</b>	<p><i>Lab Manual to Accompany Java Software Solutions.</i> Pearson Education, Inc, 2006. <a href="http://www.phschool.com">http://www.phschool.com</a></p> <p>College Board. <i>AP GridWorld Case Study.</i> New York: College Entrance Examination Board, 2006. <a href="http://apcentral.collegeboard.com">http://apcentral.collegeboard.com</a></p> <p>Daviduck, Brent. <i>Introduction to Programming in C++: Algorithms, Flowcharts, and Pseudocode</i> <a href="http://www.allclearonline.com/applications/DocumentLibraryManager/upload/program_intro.pdf">http://www.allclearonline.com/applications/DocumentLibraryManager/upload/program_intro.pdf</a></p> <p>Horstmann, Kay. <i>Big Java.</i> Hoboken, N.J.: Wiley, 2002.</p> <p>JCreator (JAVA IDE) - <a href="http://www.jcreator.com/">http://www.jcreator.com/</a></p> <p>Lewis, John, William Loftus, and Cara Cocking. <i>Java Software Solutions for AP Computer Science.</i> Second Edition. New York, New York: Pearson Education, Inc, 2007. <a href="http://www.phschool.com/advanced/course_match/cm_computer_science1.html">http://www.phschool.com/advanced/course_match/cm_computer_science1.html</a></p> <p>Sun JAVA JDK - <a href="http://java.sun.com/javase/downloads/index.jsp">http://java.sun.com/javase/downloads/index.jsp</a></p>

Syllabus at a Glance	General Topics	Sem/Week
	Introduction to Computer Systems and the Programming Environment Introduction to Objects and Primitive Data Conditional and Repetition Methods and Classes	Sem1 1 Sem1 3 Sem1 6 Sem1 10
Advanced Programming Structures AP GridWorld Case Study Responsible Use of Computer Systems Getting Ready for the AP Exam Summative Final Project	Sem2 1 Sem2 8 Sem2 12 Sem2 13 Sem2 14	

## Correlation to AP Topic Outline - Computer Science A

### I. Object-Oriented Program Design

#### A. Program design

1. Read and understand a problem description, purpose and goals	Units 1, 3, 4
2. Apply data abstraction and encapsulation	2, 4
3. Read and understand class specifications and relationships among the class ("is-a," "has-a" relationships)	4
4. Understand and implement a given class hierarchy	4
5. Identify reusable components from existing code using classes and class libraries.	2, 4

#### B. Class design

1. Design and implement a class.	4
2. Design an interface (AB Only)	4
3. Choose appropriate data representation and algorithms.	3, 4
4. Apply functional decomposition.	4
5. Extend a given class using inheritance.	4

### II. Program Implementation

#### A. Implementation technique

1. Methodology	
a. Object-oriented development	Units 2
b. Top-down development	3
c. Encapsulation and information hiding	4
d. Procedural abstraction	4

#### B. Programming constructs

1. Primitive types vs. objects	2
2. Declaration	
a. Constant declarations	2
b. Variable declarations	2
c. Class declarations	4
d. Interface declarations	4
e. Method declarations	4
f. Parameter declarations	4
3. Console output (System.out.print/println)	2
4. Control	
a. Methods	4
b. Sequential	3
c. Conditional	3
d. Iteration	3
e. Recursion	5
<b>C. Java library classes (included in the A-level Java Subset)</b>	
Object	4
Comparable	4
Integer	2
Double	2
String	2
Math	2
Random	2
ArrayList	5
<b>III. Program Analysis</b>	
<b>A. Testing</b>	
1. Test classes and libraries in isolation.	Units 4
2. Identify boundary cases and generate appropriate test data	4
3. Perform integration testing.	4
<b>B. Debugging</b>	
1. Categorize errors: compile-time, run-time, logic.	1, 2, 4, 5
2. Identify and correct errors	1, 2, 4, 5
3. Employ techniques such as using a debugger, adding extra output statements, or hand-tracing code.	1, 4
<b>C. Understand and modify existing code</b>	1-6
<b>D. Extend existing code using inheritance</b>	4
<b>E. Understand error handling</b>	
1. Understand runtime exceptions	4
2. AB only	
<b>F. Reason about programs</b>	
1. Pre-conditions and post-conditions	4
2. Assertions	4

<b>G. Analysis of algorithms</b>	
1. Informal comparisons of running time	5
2. Exact calculations of statement execution counts	5
3. AB only	N/A
4. AB only	N/A
<b>H. Numerical representations and limits</b>	
1. Representation of numbers in different bases	1
2. Limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error)	2, 3
<b>IV. Standard Data Structures</b>	
A. Simple data types (int, boolean, double)	Units 2
B. Classes	2, 4
C. One-dimensional arrays	5
D. AB only	N/A
E. AB only	N/A
F. AB only	N/A
G. AB only	N/A
H. AB only	N/A
I. AB only	N/A
J. AB only	N/A
K. AB only	N/A
L. AB only	N/A
<b>V. Standard Algorithms</b>	
<b>A. Operations on AB-level data</b>	
1. Traversals	Units 5
2. Insertions	5
3. Deletions	5
4. AB only	N/A
<b>B. Searching</b>	
1. Sequential	5
2. Binary	5
3. AB only	N/A
<b>C. Sorting</b>	
1. Selection	5
2. Insertion	5
3. Mergesort	5
4. AB only	N/A
5. AB only	N/A
<b>VI. Computing in Context</b>	

<b>A. Major hardware components</b>	
1. Primary and secondary memory	Units 1
2. Processors	1
3. Peripherals	1
<b>B. System software</b>	
1. Language translators/compilers	1
2. Virtual machines	1
3. Operating systems	1
<b>C. Types of systems</b>	
1. Single-user systems	1
2. Networks	1
<b>D. Responsible use of computer systems</b>	
1. System reliability	1
2. Privacy	7
3. Legal issues and intellectual property	7
4. Social and ethical ramifications of computer use	7

Unit #	Starting Week	Unit Title References/Readings - - - Unit Topics - - - Unit Evaluations	Unit Comments
<b>1</b>	<b>Sem 1, Week 1, 12 Days</b>	<b>UNIT 1: Introduction to Computer Systems and the Programming Environment</b>  <b>Resources:</b> <ul style="list-style-type: none"> <li>• <a href="http://www.jcreator.com/">http://www.jcreator.com/</a></li> <li>• <a href="http://java.sun.com/javase/downloads/index.jsp">http://java.sun.com/javase/downloads/index.jsp</a></li> <li>• Lewis, Loftus, Cocking: Chapter 1</li> </ul>	This is an introduction to computer systems and the programming environment. Students learn about the major hardware components of a computer system (system processor

	<p><b><u>Objectives</u></b></p> <ul style="list-style-type: none"> <li>• Describe the relationship between hardware and software.</li> <li>• Define various types of software and how they are used.</li> <li>• Identify basic computer hardware and explain what it does.</li> <li>• Explain how the hardware components execute programs and manage data.</li> <li>• Describe how computers are connected together into networks to share information.</li> <li>• Explain the importance of the Internet and the World Wide Web.</li> <li>• Introduce the Java Programming Language.</li> <li>• Describe the steps involved in program compilation and execution.</li> <li>• Introduce graphics and their representations.</li> <li>•</li> <li>• Downloading and installing Java software</li> <li>• Downloading and Installing an Integrated Development Environment (IDE)</li> <li>• Setting up the Java programming workspace</li> <li>• Translators (Compilers and interpreters)</li> <li>• The Java Virtual Machine (JVM)</li> <li>• Compiling and executing Java applications and applets</li> <li>• Java input/output issues</li> <li>• Fundamental program text output with <b>print</b> and <b>println</b></li> <li>• Java compile errors</li> </ul>	<p>and memory), system software (operating systems), types of systems, networks, the Internet, and how hardware and software work together.</p> <p>Students will learn how to use the Java JDK and an IDE to write programs. The lab assignment involves writing working programs. The intent at this stage is to learn the mechanics of working with the software and clearly understanding the compiling and executing process. In particular students will learn the difference between compiler and interpreter translators. Once the different translators are clear, students then learn how Java uses both translators.</p>
	<p><b><u>Evaluations</u></b></p> <p><b>Introduction/Binary Numbers:</b> Reading: Pgs. 2 - 9, Self Review: 1.1 – 1.5, Multiple Choice: 1.1 - 4, True/False: 1.1, Short Answer: 1.2 – 1.3</p> <p><b>Hardware Components and Networks:</b> Reading: Pgs. 10 – 24, Self Review: 1.6 – 1.12, Multiple Choice: 1.5, True/False: 1.2 – 1.6, Short Answer: 1.1, 1.4 – 1.6</p> <p><b>Programming and Programming Languages:</b> Self Review: 1.13 – 1.17, Multiple Choice: 1.6 – 1.8, True/False: 1.7 – 1.10, Short Answer: 1.7 – 1.9, Programming Projects: 1.1 – 1.6.</p> <p><b>Graphics:</b> Reading: Pgs. 42 - 44, Self Review: 1.18, Short Answer: 1.10</p> <p><b>Lab Assignments:</b> Chapter 1: Printing strings, documentation, identifiers, and syntax errors</p> <p><b>GridWorld Case Study Part I</b></p>	<p>Students will be introduced to the GridWorld Case Study Part I</p>

		Unit test	
2	Sem 1, Week 3, 15 Days	<b>UNIT 2: Introduction to Objects and Primitive Data</b> <b>Resources:</b> <ul style="list-style-type: none"> <li>Lewis, Loftus, Cocking: Chapter 2</li> </ul>	<p>Students will learn the format of a proper Java program, declare variables, and discuss what happens at each point in declaring an object, creating an instance of that object, assigning vs. changing the state of the object and how that differs from what happens with primitive data.</p> <p>Students will begin to do mathematical calculations and should come to understand both the mathematical operators and their precedence.</p> <p>Students will use simple input/output, primitive data types, the String class, arithmetic expressions, and random number generation while applying data encapsulation and data abstraction.</p>
		<b>Objectives</b> <ul style="list-style-type: none"> <li>Define the difference between primitive data and objects.</li> <li>Declare and use variables.</li> <li>Perform mathematical computations.</li> <li>Create objects and use them.</li> <li>Explore the difference between a Java application and a Java applet.</li> <li>Create graphical programs that draw shapes.</li> </ul> <b>Lab Assignments</b> <u>Lab 1</u> Phase I: Names and Places(pg. 9), Two Meanings of Plus(pg. 11 – 12) Phase II: Table of Student Grades (pg. 10)  <u>Lab 2</u> Phase I: Area and Circumference of a Circle(pg. 16 – 17), Painting a room(pg. 18), Lab Grades (pg. 20 - 21) Phase II: Ideal Weight (pg. 19), Base Conversion (22 – 23)  <u>Lab 3</u> Phase I: Working with Strings (pg. 28) , Computing Distance (pg. 30) Phase II: Rolling Dice (pg. 31)  <u>Lab 4</u> Phase I: Drawing Shapes(pg. 35 - 36), The Java Coordinate System (pg. 37 – 38), Colors in Java (pg. 41 – 42) Phase II: Drawing a face(pg. 39), Creating a Pie Chart(pg. 40)  <b>Evaluations</b> <b>Introduction to Objects/Basic Coding:</b> Reading: Pgs. 58 – 63, Self Review 2.1 – 2.2, Multiple Choice 2.1, True/False 2.1	

		<p><b>Strings, Variables and Assignments, Primitive Data Types:</b> Reading: Pgs. 63 – 79, Self Review 2.4 – 2.14, Multiple Choice 2.2 – 2.6, True/False 2.2 – 2.6, Short Answer 2.1 – 2.9, Programming Projects 2.1 – 2.7</p> <p><b>Applets/Graphics</b> Reading: Pgs. 100 – 108, Self Review 2.19, Programming Projects 2.14 – 2.15</p> <p><b>Unit test</b></p>	
3	Sem 1, Week 6, 17 Days	<p><b>UNIT 3: Conditional and Repetition</b></p> <p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• Lewis, Loftus, Cocking: Chapter 3</li> <li>• Daviduck, Brent. <i>Introduction to Programming in C++: Algorithms, Flowcharts, and Pseudocode</i> <a href="http://www.allclearonline.com/applications/DocumentLibraryManager/upload/program_intro.pdf">http://www.allclearonline.com/applications/DocumentLibraryManager/upload/program_intro.pdf</a></li> </ul>	<p>Students will become familiar with pseudocode and flow chart structures. The use of flow charts and syntax diagrams are used throughout the primary text and are integrated as a visual representation of the processes for students.</p>
		<p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>• Discuss basic program development steps.</li> <li>• Define the flow of control through a program.</li> <li>• Learn to use if statements.</li> <li>• Define expressions that let us make complex decisions.</li> <li>• Learn to use while and for statements.</li> <li>• Use conditionals and loops to draw graphics.</li> </ul>	<p>Students will become familiar with both decision (if, if-else, switch) and looping structures (for, while, do while) and the process of decomposing a problem to decide where these structures are necessary.</p>
		<p><b>Lab Assignments:</b></p> <p><u>Lab 1</u> Phase I: Computing a Raise (pg. 45), Rock Paper Scissors(pg. 48), Date Validation(pg. 49) Phase II: Computing a Charge Account Statement(pg. 46), Activities at Lake LazyDays (pg. 47)</p> <p><u>Lab 2</u> Phase I: Processing Grades (pg. 51 – 52)</p> <p><u>Lab 3</u> Phase I: Counting and Looping (pg. 55), Powers of 2 (pg. 56), A Guessing Game (pg. 58) Phase II: Factorials (pg. 57), More Guessing (pg. 59)</p>	

		<p><u>Lab 4</u> Phase I: Election Day (pg. 60), Counting Characters (pg. 63 – 64) Phase II: Finding Maximum/Minimum Values (pg. 61 – 62)</p> <p><u>Lab 5</u> Phase I: A Rainbow Applet (pg. 65)</p> <p><b><u>Evaluations</u></b></p> <p><b>Control Flow and Basic Decision Structures:</b> Reading: Pgs. 122 – 135, Self Review: 3.1 – 3.8, Multiple Choice: 3.1 – 3.2, True/False: 3.1 – 3.3, Short Answer: 3.1 – 3.6, Programming Projects: 3.1 – 3.2</p> <p><b>Compound Boolean Statements:</b> Reading: Pgs. 135 – 142, Self Review: 3.9 – 3.11, Multiple Choice: 3.3 – 3.9, True/False: 3.4 – 3.6</p> <p><b>Repetition Structures, The while statement:</b> Reading: Pgs. 142 – 156, Self Review: 3.12, Multiple Choice: 3.7, True/False: 3.8, Short Answer: 3.7, 3.10 – 3.13, Programming Projects: 3.3, 3.4, 3.6, 3.11, 3.15, 3.16</p> <p><b>Repetition Structures, The do and for statements:</b> Reading: Pgs. 156 – 166, Self Review: 3.13, Multiple Choice: 3.4 – 3.6, 3.8, True/False: 3.7, 3.9, Short Answer: 3.8, 3.9, 3.14 – 3.19, Programming Projects: 3.5, 3.7 – 3.10, 3.12 – 3.14</p> <p><b>Drawing with Loops and Conditionals</b> Reading: Pgs. 168 – 174</p> <p><b>Unit test</b></p>	
<p><b>4</b></p>	<p><b>Sem 1, Week 10, 45 Days</b></p>	<p><b>UNIT 4: Methods and Classes</b></p> <p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>Lewis, Loftus, Cocking: Chapters 4, 5, and 7</li> </ul> <p><b><u>Objectives</u></b></p> <ul style="list-style-type: none"> <li>Define classes that act like blue-prints for new objects, made of variables and methods.</li> <li>Explain encapsulation and Java modifiers.</li> <li>Explore the details of method declarations.</li> <li>Review method invocation and parameter passing.</li> <li>Explain and use method overloading.</li> <li>Learn to divide complicated methods into simpler, supporting methods.</li> </ul>	<p>Students learn how to create their own classes by defining objects. Proper method and class structure is emphasized.</p> <p>This unit explores some of the concepts of how to break code into usable pieces as well as some of the important features of</p>

		<ul style="list-style-type: none"> <li>• Describe relationships between objects.</li> <li>• Create graphics based objects.</li> <li>• Define reference aliases.</li> <li>• Explore passing object references as parameters.</li> <li>• Learn to use the static modifier.</li> <li>• Define formal interfaces and their class implementations.</li> <li>• Define nested classes and inner classes</li> <li>• Learn about basic graphical user interfaces.</li> <li>• Derive new classes from existing ones.</li> <li>• Explain how inheritance supports software reuse.</li> <li>• Add and modify methods in child classes.</li> <li>• Discuss how to design class hierarchies.</li> <li>• Define polymorphism and how it can be done.</li> <li>• Discuss the use of inheritance in Java GUI framework.</li> <li>• Examine and use the GUI component class hierarchy.</li> </ul>	<p>designing abstract data types and data storage classes. It then takes a more in depth look at objects and the mechanisms that go into creating a robust system. Students will spend ample time decomposing everyday objects into their properties and behaviors and identifying the difference between the two. Interfaces (Comparable), polymorphism and inheritance are major concepts in this unit.</p>
		<p><b><u>Lab Assignments</u></b></p> <p><u>Lab 1</u> Phase I: Using the Coin Class (pgs. 68 – 69) Phase II: Modifying the Coin Class (pg. 70)</p> <p><u>Lab 2</u> Phase I: A Bank Account Class (pgs. 71 – 73)</p> <p><u>Lab 3</u> Phase I: Tracking Grades (pgs. 74 – 76) Phase II: Band Booster (pg. 77)</p> <p><u>Lab 4</u> Phase I: Representing Names (pg. 78)</p> <p><u>Lab 5</u> Phase I: Drawing Circles (pg. 79)</p> <p><u>Lab 6</u> Phase I: Tracing References (pg. 81 – 83), Changing People (pg. 84 – 85)</p> <p><u>Lab 7</u> Phase I: A Modified MiniQuiz Class (pg. 87 - 89), Bank Account Transfers (pg. 90 – 91), Counting Transactions (pg. 92 – 93) Phase II: Using the Comparable Interface (pg. 86)</p> <p><u>Lab 8</u></p>	<p>Students learn how to use input and output streams. Topics include exceptions, standard I/O, and reading from and writing to text files.</p>

Phase I: Modifying EvenOdd.java (pg. 94)  
Phase II: A Pay Check Program (pg. 95)

Lab 9

Phase I: Voting with Buttons (pg. 96 – 97), Calculating Body Mass Index (pg. 98 – 100)

Lab 10

Phase I: Exploring Inheritance (pg. 130-133), A Sorted Integer List (Pg. 134 – 135), Another Type of Employee of the Firm (pg. 138 – 144)

Phase II: Test Questions (pg. 136 – 137), Overriding the equals Method (pg. 145 – 146)

Lab 11

Phase I: Drawing Circles with MouseClicks (pg. 147 – 151), More Listeners (pg. 153 – 155)

Phase II: Moving Circles with the Mouse (pg. 152)

**Evaluations**

**Anatomy of Classes and Methods:**

Reading: Pgs. 190 – 210, Self Review: 4.1 – 4.7, Multiple Choice: 4.1 – 4.3, True/False: 4.1 – 4.4, Short Answer: 4.1 – 4.13, Programming Projects: 4.1 – 4.3, 4.6

**Method Overloading:**

Reading: Pgs. 210 – 214, Self Review: 4.8 – 4.9, Multiple Choice: 4.4 – 4.6, True/False: 4.5, Short Answer: 4.14 – 4.18, Programming Projects: 4.8

**Method Decomposition:**

Reading: Pgs. 214 – 219, Self Review: 4.10, Multiple Choice: 4.7 – 4.10, Short Answer: 4.19 – 4.22, Programming Projects: 4.4

**Object Relationships:**

Reading: Pgs. 219 – 228, Self Review: 4.11 – 4.12, True/False: 4.6 – 4.10, Programming Projects: 4.5, 4.7

**Applets and Graphics:**

Reading: Pgs. 231 – 235, Self Review: 4.13, Short Answer: 4.23 – 4.25, Programming Projects: 4.9 – 4.10

**References/Passing Parameters:**

Reading: Pgs. 248 – 263, Self Review: 5.1 – 5.5, Multiple Choice: 5.1 – 5.3, True/False: 5.1 – 5.8, Short Answer: 5.1 – 5.2, Programming Projects: 5.1

**Interfaces:**

Reading: Pgs. 263 – 273, Self Review: 5.6 – 5.7, Multiple Choice: 5.4 – 5.10, True/False: 5.9 – 5.10, Short Answer: 5.2 – 5.6, Programming Projects: 5.2 – 5.7

		<p><b>Dialog Boxes:</b> Reading: Pgs. 277 – 279, Self Review: 5.8, Programming Projects: 5.8 – 5.9</p> <p><b>Graphical User Interfaces:</b> Reading: Pgs. 280 – 288, Self Review: 5.9 – 5.10, Programming Projects: 5.10</p> <p><b>Inheritance:</b> Reading: Pgs. 382 - 423, Self Review: 7.1 – 7.13, Multiple Choice: 7.1 – 7.10, True/False: 7.1 – 7.10, Short Answer: 7.1 – 7.5, Programming Projects: 7.1 – 7.5</p> <p><b>Mouse Events:</b> Reading: Pgs.428 - 442, Short Answer: 7.6 – 7.7, Programming Projects: 7.6 – 7.8</p> <p><b>Unit test (Divided into three separate tests)</b></p>	
<p><b>5</b></p>	<p><b>Sem 2, Week 1, 36 Days</b></p>	<p><b>UNIT 5: Advanced Programming Structures</b></p> <p><b>Resources:</b> Lewis, Loftus, Cocking: Chapters 6 and 8</p> <p><u><b>Objectives</b></u></p> <ul style="list-style-type: none"> <li>• Define and Use Arrays</li> <li>• Describe how arrays and array elements are passed as parameters.</li> <li>• Explore how arrays and other objects can be combined to manage complex information.</li> <li>• Explore searching and sorting with arrays.</li> <li>• Learn to use multidimensional arrays.</li> <li>• Examine the ArrayList class.</li> <li>• Explain the underlying ideas of recursion.</li> <li>• Examine recursive methods and processing steps.</li> <li>• Define infinite recursion and discuss ways to avoid it.</li> <li>• Explain when recursion should and should not be used.</li> <li>• Demonstrate the use of recursion to solve problems.</li> <li>• Examine the use of recursion in sorting.</li> </ul>	<p>This unit is designed to give students an introduction to information stored in indexed data structures. Topics explored include built-in arrays, arrays of objects, and the use of the Java Collection Class ArrayList. Sorting algorithms (Insertion Sort, Selection Sort, and Merge Sort) are also explored here. This unit also covers recursion through examples of some classic problems as well as through graphical drawing of fractals.</p> <p>Students will learn to develop and select</p>

		<p><b><u>Lab Assignments</u></b></p> <p><u>Lab 1</u> Phase I: Tracking Sales(pg. 102), Searching and Sorting In an Integer List (pg. 105 – 108), Adding to and Removing From an Integer List (pg. 109) Phase II: Grading Quizzes(pg. 103), Magic Squares (pg. 110 – 113)</p> <p><u>Lab 2</u> Phase I: A Shopping Cart(pg. 114 – 115)</p> <p><u>Lab 3</u> Phase I: A Shopping Cart using the ArrayList Class (pg. 117 – 118)</p> <p><u>Lab 4</u> Phase I: Drawing Circles (Again) (pg. 119 – 120), A Polygon Person (pg. 121 – 122)</p> <p><u>Lab 5</u> Phase I: Adding Buttons to StyleGUI.java (pg. 123 – 125), An Array of Radio Buttons (pg. 126 – 128)</p> <p><u>Lab 6:</u> Phase I: Computing Powers(pg. 225), Counting and Summing Digits in an Integer(pg. 226 – 227) Phase II: Base Conversion (pg. 228 – 229) , Efficient Computation of Fibonacci Numbers (pg. 230-231)</p> <p><u>Lab 7:</u> Phase I: Palindromes (pg. 232) Phase II: Printing a String Backwards (pg. 233)</p> <p><u>Lab 8:</u> Phase I: Recursive Sequential Search (pg. 234 – 237), Recursive Binary Search (pg. 238 – 242) Phase II: Recursive Array Processing, A List of Employees (pg. 243 – 246)</p> <p><u>Lab 9:</u> Phase I: Sierpinski Triangles (pg. 247) Phase II: Modifying the Koch Snowflake (pg. 248 – 252)</p> <p><b><u>Evaluations</u></b></p> <p><b>One and Two Dimensional Primitive Arrays:</b> Reading: Pgs. 298 – 308, 333 - 337, Self Review: 6.1 – 6.5, Multiple Choice: 6.1 – 6.3, True/False: 6.1 – 6.5, Short Answer: 6.1 – 6.3, Programming Projects: 6.1 – 6.5</p> <p><b>Arrays of Objects:</b> Reading: Pgs. 308 – 333, Self Review: 6.6 – 6.11, Multiple Choice: 6.4 – 6.6, True/False: 6.6 – 6.10, Short Answer: 6.4 – 6.5, Programming Projects: 6.6 – 6.11</p>	<p>appropriate algorithms and data structures to solve problems.</p>
--	--	--	--

		<p><b>ArrayLists:</b> Reading: Pgs. 337 - 342, Self Review: 6.12, Multiple Choice: 6.7 – 6.10, Short Answer: 6.6 - 6.7, Programming Projects: 6.12 – 6.14</p> <p><b>Arrays and Graphics:</b> Reading: Pgs. 358 - 361, Self Review: 6.13, Short Answer: 6.8, Programming Projects: 6.15 – 6.16</p> <p><b>Buttons:</b> Reading: Pgs. 361 - 369, Self Review: 6.14</p> <p><b>Basic Recursion:</b> Readings: pgs. 454 – 464, Self Review: 8.1 – 8.6, Multiple Choice: 8.1 – 8.5, True/False: 8.1 – 8.5, Short Answer: 8.1 – 8.6, Programming Projects: 8.1 – 8.2</p> <p><b>Recursion on Strings:</b> Readings: pgs. 465 – 469, Self Review: 8.7, Multiple Choice: 8.6, True/False: 8.6 – 8.7, Short Answer: 8.7 – 8.8, Programming Projects: 8.3 – 8.8</p> <p><b>Recursion on Lists:</b> Readings: pgs. 469 – 476, Self Review: 8.8 – 8.9, Multiple Choice: 8.7 – 8.10, True/False: 8.8 – 8.10, Short Answer: 8.9 – 8.11, Programming Projects: 8.9 – 8.10</p> <p><b>Fractals:</b> Readings: pgs. 493 – 503, Self Review: 8.10, Short Answer: 8.12, Programming Projects: 8.11</p> <p><b>Unit test (Broken into 2 tests)</b></p>	
<p><b>6</b></p>	<p><b>Sem 2, Week 8, 15 Days</b></p>	<p><b>UNIT 6: AP GridWorld Case Study</b></p> <p><b>Resources:</b> AP GridWorld Case Study</p> <hr/> <p><b>Objectives</b></p> <ul style="list-style-type: none"> <li>Students will work through Parts 2, 3, and 4 of the AP GridWorld Case Study.</li> </ul> <p><b>Evaluation</b></p> <ul style="list-style-type: none"> <li><b>Unit Test Covering Parts 2, 3, and 4 of the GridWorld Case Study</b></li> </ul>	<p>Students will work through Parts 2, 3, and 4 of the AP GridWorld Case Study</p>

<b>7</b>	<b>Sem 2, Week 12, 5 Days</b>	<b>UNIT 7: Responsible Use of Computer Systems</b>  <b>Resources:</b> Lewis, Loftus, Cocking: Appendix E (Responsible Use of Computer Systems)	Students explore how computers have affected society. Topics include system reliability, privacy, legal issues and intellectual property, and the social and ethical ramifications of computer use (improving the effectiveness of email, viruses and worms, protecting intellectual property, ensuring reliability of software, privacy, and fair information practices).
		<b>Objectives</b> <ul style="list-style-type: none"> <li>• Students will recognize the ethical and social implications of computer use.</li> </ul> <b>Evaluation</b> <ul style="list-style-type: none"> <li>• <b>Computer Careers and Ethics – Research Paper</b></li> <li>• <b>Unit Test</b></li> </ul>	
<b>8</b>	<b>Sem 2, Week 13, 10 Days</b>	<b>UNIT 8: Getting Ready for the AP Exam</b>  <b>Resources:</b> AP GridWorld Case Study Supplemental handouts – multiple-choice and free-response questions	Students work through sample multiple-choice questions and free-response questions.
<b>9</b>	<b>Sem 2, Week 15, 15 Days</b>	<b>UNIT 9: Summative Final Project</b>  <b>Evaluation:</b> Students will make a presentation detailing the design and implementation of their project	Students will work though a final project that evaluates their ability to work through the development of a large-scale program they have proposed, designed and written. Tasks include writing a program proposal, planning the program using pseudocode/flowcharts, creating a user's manual, writing program code, and creating a limitations document.

# **Student / Teacher / Parent AP Computer Science A Agreement**

**Student:** I \_\_\_\_\_ (print name), have read and understand the classroom policies as outlined in the course syllabus. I understand that it is my responsibility to uphold all policies and begin my pursuit of a high-level understanding of Java programming. I will put forth the effort needed to successful obtain a high-level of understanding and achievement of AP Computer Science A topics.

Student Signature \_\_\_\_\_

Date \_\_\_\_\_

**Parent/Guardian:** I have read, understand, and acknowledge what is expected of my child in this AP Computer Science A course and agree to support him/her throughout the school year.

Printed Parent Name \_\_\_\_\_

Parent Signature \_\_\_\_\_

Date \_\_\_\_\_

**Teacher:** I, the AP Computer Science Instructor have read, understood, and acknowledged what is expected of the student in this AP Computer Science A course and agree to provide them with instruction which creates a safe and successful learning environment. I intend to parallel the hard work needed from the students.

Teacher Signature \_\_\_\_\_  
(Signed after return of Student & Parent Signatures)

Date \_\_\_\_\_