



## Gifted LearningLinks Program Course Syllabus

**Instructor name: B. Fox**

**E-mail address: xxx@xxx.xxx**

**Course Title: AP Computer Science A**

**Session Date:** 2010-2011, *monthly enrollments*

### **Course Description:**

AP® Computer Science A is a two-semester course that is the equivalent to a first-semester college introductory course in Computer Science. It teaches object-oriented programming methodology using Java. The course teaches students how to design, code, debug, and implement practical programs. Design includes ideas such as well-documented, reusable, and adaptable code. Coding concentrates on higher level skills involving problem solving and algorithmic development. Technology terminology and concepts are presented to help the students understand basic hardware, software, and operating system components.

This course is targeted toward students who are considering taking college courses in such areas as Computer Science, Information Technology, or Engineering. Students will write OOP (object-oriented) programs utilizing classes, objects, methods, class inheritance, and polymorphism using the standard Java library classes from the AP® Java subset. They will also explore the AP® GridWorld Case Study and be expected to take the AP® Computer Science A Exam at the end of the year.

### **Student Responsibilities**

Students must successfully complete either Honors Java or Honors C++. They will be required to install a Java IDE at home so they can complete assignments and make up work. Students must expect to spend at least 4-6 hours per week for 9 months to complete assignments and study.

### **Teacher Responsibilities**

I believe the successful AP Computer Science teacher has three main responsibilities:

- 1) present new or complex topics in a way that students can understand
- 2) encourage the students to be self-motivated to code and learn with minimal direction
- 3) assist the students when they run into problems

I encourage the students to ask each other for help (for both debugging and when implementing new coding practices) because this helps teach them the problem-solving skills necessary to be successful in Computer Science. As much as possible, I also try to offer open-ended assignments or options such that students can creatively implement new knowledge in their own unique way.

### **Assessment**

The majority of class time will be spent typing code, so programs will be the largest individual factor contributing to the final grade. Self-assessment and teacher-assessment rubrics will be used for scoring. Students will be required to document their design decisions and self-document

their code for full-credit using comments. Flow charts, UML notation and other design specification documents will also be produced. Homework will be used to introduce new concepts, review current knowledge, and complete programs outside of class. Quizzes and tests will be used to maintain high accountability on reading assignments and to prepare the students for the AP® Computer Science A Exam. After sufficient in-class practice, students will also be asked to complete AP® free response questions from previous tests and sample multiple-choice questions.

### **Facilities**

This is a Distance Learning course taught using a Facilitated Learning Model. In other words, the teacher sets up the course so that the students proceed at their own pace with regular check points with the teacher. Therefore, students will be learning from either a home computer or a school computer (some take this class during the school day). They must have the Java compiler and libraries available and will typically use an IDE such as BlueJ, drJava, or Eclipse.

### **Texts**

#### **Primary Textbook:**

Lambert, Kenneth, and Martin Osborne. Fundamentals of Java: AP\* Computer Science Essentials for the A & AB Exams, Third Edition. Boston: Thomson Course Technology, 2007. ISBN # 0-619-26723-2

#### **Secondary Textbooks:**

Teukolsky, Roselyn. Barron's AP Computer Science, 2007-2008: Levels A and AB (Barron's How to Prepare for the AP Computer Science Advanced Placement Examination) Barron's Educational Series, 4<sup>th</sup> Edition, Roselyn A Teukolsky. ISBN-13: 978-0-7641-3709-9 or 978-0-7641-9350-7 (with CD).

#### **Additional Printed Resources:**

Litvin, Maria and Litvin, Gary. *Java Methods A & AB: Object Oriented Programming and Data Structures*. Skylight Publishing, 2006. <http://www.skylight.com>

College Board. AP® GridWorld Case Study. New York: College Entrance Examination Board, 2006

AP Central®: Computer Science A Quick Reference Guide

Guzdial, M. and Ericson, B. *Introduction to Computing and Programming with Java: A Multimedia Approach*

Java Software Solutions for AP Computer Science A, 2nd Edition, J. Lewis, W. Loftus, and C. Cocking, 2007, Prentice Hall.

#### **Additional Online Resources:**

<http://www.bluepelicanjava.com/>

Java 5.0 or higher

<http://java.sun.com/j2se/1.5.0/> or higher

DrJava: <http://drjava.org>

BlueJ: [www.bluej.org](http://www.bluej.org)

Teacher Resources: <http://www.gwhs.warren.k12.ky.us/pages/klee/approg.html>

***Note:** While a staff member (which includes but is not limited to instructors, teaching assistants, residential staff, office staff) may use or refer students to third-party web sites for instructional purposes, s/he is required to review thoroughly any such web sites for inappropriate content before referencing them. This includes clicking on all links contained in any such web site, reviewing the materials contained on every page within a web site, and ensuring that the web site does not provide links to other inappropriate web sites. If there is any doubt about whether something is appropriate, a staff member is instructed to err on the side of caution and not use the web site or check with a CTD supervisor. If a referenced third-party web site is later determined to contain inappropriate content, the staff member may be subject to discipline.*

### Schedule:

#### Course Outline [C2]

Following are the abbreviations for the referenced materials:

FJ – Fundamentals of Java

GW – GridWorld Case Study

BA – Barron’s AP Computer Science

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
<b>0</b> (0)	<b>Prerequisites</b> <ul style="list-style-type: none"><li>Successful completion of Programming I, Honors Java, or Honors C++ which includes:<ul style="list-style-type: none"><li>familiarity with Object Oriented terminology</li><li>introduction to variables, loops, logic, parameters, and other common usages</li><li>coding in programming language such as C++ or ActionScript</li></ul></li></ul>	<b>Resources:</b> <ul style="list-style-type: none"><li>Alice</li><li>Bloodshed Dev C++ IDE</li></ul>
<b>1</b> (0-2)	<b>Java Basics</b> <b>Topics: [C3] [C8] [C9]</b> <ul style="list-style-type: none"><li>Computer Basics</li><li>Binary Representation</li><li>Java Basics</li><li>Using the Compiler</li><li>Input and Output</li></ul> <b>Objectives:</b> <ul style="list-style-type: none"><li>Understand computer terminology</li><li>Understand computer ethics such as acceptable use policies, copyright,</li></ul>	<b>Resources:</b> <ul style="list-style-type: none"><li>FJ: Chapters 1 &amp; 2</li><li>BA: Chapter 1</li></ul> <b>Assignments:</b> Written: <ul style="list-style-type: none"><li>Number conversion worksheet</li><li>Computer Hardware Forum</li><li>FJ: Project 1-1Forum</li><li>Forum on ACM Code of Ethics</li></ul> Labs (FJ):

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p>intellectual property, freeware, shareware, downloading</p> <ul style="list-style-type: none"> <li>• Understand how the different parts of a computer work together</li> <li>• Explore operating systems</li> <li>• To convert from and to the binary, decimal, and hexadecimal number systems</li> <li>• Understand compiler terminology</li> <li>• Edit, compile, and run a simple Java program</li> <li>• Use the Scanner class to do input</li> <li>• Use System.out method of the Scanner class to format output with print and println</li> </ul>	<ul style="list-style-type: none"> <li>• HelloWorld (Ex 2.1, pg 36)</li> <li>• Temperature (Ex 2.2, pg 42)</li> <li>• Address (P2-5, pg 55)</li> <li>• Momentum (P2-6, pg 55)</li> <li>• Windows/linux system exploration to identify hardware and software</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapters 1 &amp; 2</li> <li>• Introduce DrJava or BlueJay</li> <li>• Do number conversions in class and as homework</li> <li>• Computer Ethics scenario discussions through Forum</li> <li>• Write simple programs that illustrate different types of input and output.</li> <li>• Unit test #1: multiple choice &amp; free response</li> </ul>
<p><b>2</b> (3-6)</p>	<p><b>Defining Variables, Arithmetic Expressions, Conditions, and Loops</b>  <b>Topics: [C3] [C4] [C5] [C6]</b></p> <ul style="list-style-type: none"> <li>• Declare and Use Variables</li> <li>• Arithmetic Expressions</li> <li>• Casting</li> <li>• Program Comments</li> <li>• Syntax, Run-time, and Logic Errors</li> <li>• Math Class</li> <li>• If and If-Else Statements</li> <li>• While Loops</li> <li>• For Loops</li> <li>• Nested Control with Break Statements</li> <li>• Reading and Writing to Text Files</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand terminology: comments, variables, data types, constants, reserved words, literals.</li> <li>• Declare and initialize variables and constants</li> <li>• Perform calculations using arithmetic expressions and order of operations.</li> <li>• Use type casting in arithmetic</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapters 3 &amp; 4</li> <li>• BA: Chapter 1</li> </ul> <p><b>Assignments:</b>  Written:</p> <ul style="list-style-type: none"> <li>• FJ: Exercises as homework</li> <li>• BA: Chapter 1 questions</li> <li>• Create UML Class Diagrams for various scenarios</li> <li>• Control Structures Forum where students design loops &amp; conditionals for a favorite activity</li> </ul> <p>Labs (FJ):</p> <ul style="list-style-type: none"> <li>• CubeSurfaceArea (P3-1, pg 103)</li> <li>• KineticEnergy (P3-3, pg 103)</li> <li>• TotalPay (P3-4, pg 103)</li> <li>• QuotientRemainder (P4-1, pg 146)</li> <li>• Rock, Paper, Scissors</li> <li>• AreaOfTriangle</li> <li>• PI (P4-5, pg 147)</li> <li>• CalculateEaster</li> <li>• Convert Celsius/Fahrenheit</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p>expressions.</p> <ul style="list-style-type: none"> <li>• Use various methods found in the Math class</li> <li>• Use the increment and decrement operators.</li> <li>• Use if and if-else statements to make choices.</li> <li>• Use while and for loops to repeat a process.</li> <li>• Construct appropriate conditions for control statements using relations operators.</li> <li>• Detect and correct common errors involving loops.</li> <li>• Concatenate data and parse numbers using integer division and modulus.</li> <li>• To write programs that read and writes data to text files.</li> </ul>	<ul style="list-style-type: none"> <li>• Binary/Hex/Decimal conversions</li> <li>• StudentGrades</li> <li>• Debugging Practice</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapters 3 &amp; 4.</li> <li>• PowerPoint Slides on Classes.</li> <li>• Assign a number of programs using variables, calculations, decision making, and loops.</li> <li>• Watch my video on creating UML Class Diagrams</li> <li>• Assign programs that teach common algorithms and present "Useful Segments of Java Code" resource</li> <li>• Allow cooperative learning by allowing students to post questions on Debugging Forum.</li> <li>• By writing numerous programs, students will get the needed practice that will help them develop: <ul style="list-style-type: none"> <li>- debugging skills</li> <li>- an understanding of logic needed to write selection and repetition structures</li> <li>- an understanding of how data types and type casting can affect mathematical calculations.</li> </ul> </li> <li>• Quizzes</li> <li>• Unit Test #2: multiple choice and free response</li> </ul>
<p><b>3</b> (7-10)</p>	<p><b>Introduction to Defining Classes</b>  <b>Topics:</b> [C4] [C5] [C6]</p> <ul style="list-style-type: none"> <li>• Create and Use Classes</li> <li>• The Internal Structure of Classes and Objects</li> <li>• The Student Class</li> <li>• The Structure and Behavior of Methods</li> <li>• Scope and Lifetime of Variables</li> <li>• toString()</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Design and implement a simple class</li> </ul>	<p><b>Resources:</b></p> <p>FJ: Chapter 5  BA: Chapter 2  BA: Chapter 5A</p> <p><b>Assignments:</b>  Written:</p> <ul style="list-style-type: none"> <li>• Handouts on Classes</li> <li>• Student (P5-1 &amp; P5-2, pg 196)</li> <li>• Rectangle (P5-8, pg 198)</li> <li>• Fraction (P5-3, pg 196)</li> <li>• Software Development Life Cycle practice</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p>from user requirements.</p> <ul style="list-style-type: none"> <li>• To apply the model/view pattern to structure program code.</li> <li>• Use visibility modifiers to make methods visible to clients and restrict access to data within a class.</li> <li>• To write constructors, mutator methods, and accessor methods for a class.</li> <li>• To understand how messages are transmitted to methods through parameters</li> <li>• Use instance variables, locals variables, and parameters appropriately.</li> <li>• Use helper methods to organize a complex task.</li> <li>• Learn how to write toString() methods and understand how they are used implicitly when an object is "printed"</li> </ul>	<ul style="list-style-type: none"> <li>• Create UML Class Diagrams for BankAccount scenario</li> <li>• People Forum - each student must choose a person with special skills and list class name, instance variables, constructors, methods, etc.</li> <li>• Library (P5-6, p. 197) with UML</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapter 5 and do FJC chapter 5 exercises.</li> <li>• PowerPoints on FJ chapter 5. Study how to create a class by writing the rectangle class.</li> <li>• Assign a different geometric shape (square, triangle, circle, sphere, cylinder, rectangular prism, cone, triangular pyramid, square pyramid, and hexagonal prism) to each student based upon their mathematic background and have them write an appropriate representation for that class.</li> <li>• Allow cooperative learning on programs that are more difficult for students.</li> <li>• Practice quiz</li> </ul>
<p><b>4</b> (11-14)</p>	<p><b>Advanced Control Statements</b> <b>Topics: [C3] [C6]</b></p> <ul style="list-style-type: none"> <li>• Logical Operators</li> <li>• Nested If Statements</li> <li>• Nested Loops</li> <li>• Testing If Statements and Loops</li> <li>• Do-While Loop</li> <li>• Switch Statement</li> <li>• Menu-Driven and Query-Driven Programs</li> <li>• Formatting Output</li> <li>• Try-Catch Statement</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Construct complex Boolean expressions</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapters 6 &amp; 7</li> <li>• BA Chapter 1</li> </ul> <p><b>Assignments:</b> Written:</p> <ul style="list-style-type: none"> <li>• Labs (FJ):</li> <li>• GuessingGame (P6-1, pg 244)</li> <li>• GradingScale (P6-3, pg 245)</li> <li>• WeightOnPlanets</li> <li>• Grid World Part 2 exercises</li> <li>• Selected Case Studies</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapters 6 &amp; 7</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p>using the logical operators.</p> <ul style="list-style-type: none"> <li>• Understand the logic of nested if statements.</li> <li>• Construct nested loops.</li> <li>• Create appropriate test cases for if statements and loops.</li> <li>• Construct query-driven and menu-driven programs.</li> <li>• Format text and numbers for output</li> <li>• Handle number format exceptions during input.</li> <li>• Introduce Case Studies</li> </ul>	<ul style="list-style-type: none"> <li>• PowerPoints over FJ chapters 6 &amp; 7 and show numerous examples of program code containing if statements, loops, and formatting.</li> <li>• Assign a large number of programs using if statements and loops using codingbat.com and PracticeIt websites.</li> <li>• Assign programs utilizing “real life” algorithms.</li> <li>• Students are asked to apply concepts learned in their math and science classes to programs that are written in this class.</li> <li>• The programs written by the students are terminal I/O user interface programs. Students need experience in writing query-driven and menu-driven programs and formatting output to improve the user interface.</li> <li>• Allow cooperative learning on programs that are more difficult for students.</li> <li>• Unit test #4: multiple choice and free response</li> </ul>
<p><b>5</b> (15-18)</p>	<p><b>Introduction to Arrays</b> <b>Topics: [C4] [C5] [C6]</b></p> <ul style="list-style-type: none"> <li>• Declare, construct, initialize and index Arrays and ArrayLists</li> <li>• java.util.List methods</li> <li>• Simple Array Manipulations Using Loops</li> <li>• Declaring and Initializing Arrays</li> <li>• Traverse, insert, delete Array &amp; ArrayList elements</li> <li>• Working with Arrays That Are Not Full</li> <li>• Parallel Arrays</li> <li>• Two-Dimensional Arrays</li> <li>• Using the Enhance For Loop</li> <li>• Passing Arrays/ArrayLists to methods</li> <li>• Java 5.0’s Generics</li> <li>• Java 5.0’s enhanced for loop</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapter 9</li> <li>• BA: Chapter 6</li> </ul> <p><b>Assignments:</b> Written:</p> <ul style="list-style-type: none"> <li>• Labs (FJ):</li> <li>• Student (P9-6, pg 347)</li> <li>• EvenOdd (P901, pg 347)</li> <li>• GreaterThanAverage (P9-2, pg 347)</li> <li>• Probability</li> <li>• GridWorld P3</li> <li>• CardShuffleAndDeal</li> <li>• MagicSquare (P9-7, pg 348)</li> <li>• PennyPitch (P9-10, pg 349)</li> <li>• Nibbles</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<ul style="list-style-type: none"> <li>• Arrays of Objects</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understand array concepts: element, index, physical size, logical size, parallel arrays, two-dimensional arrays.</li> <li>• Declare and initialize one-dimensional arrays.</li> <li>• Declare and initialize two-dimensional arrays.</li> <li>• Using initializer lists with arrays.</li> <li>• Manipulate arrays with loops, including the enhanced for loop.</li> <li>• Write methods to manipulate arrays.</li> <li>• Create parallel arrays.</li> <li>• Work with arrays of both primitive data types and arrays of objects.</li> <li>• Use arrays where the physical and logical size are different.</li> <li>• Understand that ArrayList implements List.</li> </ul>	<p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapter 9</li> <li>• PowerPoint over FJ chapter 9 and show sample program code containing one and two-dimensional arrays.</li> <li>• Assign a number of programs using arrays so that students practice manipulating arrays using loops using book, codingbat, and PracticeIt</li> <li>• Students need to remember that the indexing of arrays begins with the number zero.</li> <li>• Unit Test #5: multiple choice and free response</li> </ul>
<p><b>6</b> (19-21)</p>	<p><b>Advanced Operations on Arrays, Strings, and ArrayLists</b></p> <p><b>Topics: [C4] [C5] [C6]</b></p> <ul style="list-style-type: none"> <li>• String Class &amp; Methods</li> <li>• Wrapper classes for Integer and Double</li> <li>• Integer.MAX_VALUE &amp; Integer.MIN_VALUE</li> <li>• Linear Search</li> <li>• Binary Search</li> <li>• Selection Sort</li> <li>• Insertion Sort</li> <li>• Bubble Sort</li> <li>• Insertions and Removals</li> </ul> <p style="padding-left: 40px;">Arrays of Objects ArrayLists</p> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• To instantiate strings.</li> <li>• Use the appropriate String accessor methods to solve a problem.</li> <li>• Write a method for searching an array.</li> <li>• Understand why a sorted array can be</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapter 11</li> <li>• BA: Chapter 6</li> </ul> <p><b>Assignments:</b></p> <p>Written:</p> <ul style="list-style-type: none"> <li>• String Worksheet</li> <li>• Search Worksheet</li> <li>• Sorting Worksheet</li> <li>• Insertions and Removal Worksheet</li> <li>• ArrayList Worksheet</li> </ul> <p>Labs (FJ):</p> <ul style="list-style-type: none"> <li>• PigLatin</li> <li>• Vowels</li> <li>• FirstLetter, FindWord</li> <li>• LinearSearch, BinarySearch</li> <li>• SelectionSort</li> <li>• InsertionSort</li> <li>• BubbleSort</li> <li>• InsertRemove</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p>searched more efficiently than an unsorted array.</p> <ul style="list-style-type: none"> <li>• Write a method to sort an array.</li> <li>• Understand the logic and be able to trace an array being searched or sorted using one of the following: Linear Search, Binary Search, Bubble Sort, Insertion Sort, Selection Sort.</li> <li>• Understand when to use a certain search or sort method and determine which one is most appropriate for the situation.</li> <li>• Write methods to perform insertions and removals at given positions in an array.</li> <li>• Understand the issues involved when working with arrays of objects.</li> <li>• To be able to declare, instantiate, and initialize an ArrayList.</li> <li>• Use the appropriate ArrayList method to solve a problem.</li> <li>• Explore wrapper classes like Integer and their built-in constants of Integer.MAX_VALUE and Integer.MIN_VALUE</li> </ul>	<ul style="list-style-type: none"> <li>• Convert to ArrayLists</li> <li>• PlayerStats</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapter 11 and do FJC chapters 11 exercises.</li> <li>• Cover FJ chapter 11 and show sample program code of String methods, ArrayList methods, searching, and sorting methods.</li> <li>• Assign a large number of programs using arrays and ArrayLists. Have students work with the different sorts and searches.</li> <li>• Students need to do a worksheet on all the String methods (especially substring, length, equals, compareTo, and indexOf) to reinforce what each method does to a String.</li> <li>• Stress to students that they need to understand how to use the following ArrayList methods: get, isEmpty, indexOf, remove, size, set, and both add methods.</li> <li>• Show students that after using the add and remove methods that the size is automatically updated.</li> <li>• Students will be given a worksheet where they must trace through different searches and sorts to help them understand the logic behind each.</li> <li>• Unit test #6: written and program</li> </ul>
<p>7 (22-24)</p>	<p><b>Recursions and Complexity Analysis</b>  <b>Topics:</b> [C4] [C5] [C6]</p> <ul style="list-style-type: none"> <li>• Recursion</li> <li>• Complexity Analysis</li> <li>• Binary Search Big-O</li> <li>• QuickSort</li> <li>• Merge Sort</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapter 12</li> <li>• BA: Chapter 7</li> </ul> <p><b>Assignments:</b>  Written:</p> <ul style="list-style-type: none"> <li>• Recursion Worksheet</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Design and implement a recursive method to solve a problem.</li> <li>• To understand the differences and similarities between iteration and recursion solutions to a problem.</li> <li>• Understand how a recursive method calls itself.</li> <li>• Perform a simple complexity analysis of an algorithm.</li> <li>• Recognize some typical big-O values to determine the most efficient method.</li> <li>• Understand the behavior of and how to use the QuickSort and Merge Sort algorithms.</li> </ul>	<ul style="list-style-type: none"> <li>• Big-O Worksheet</li> <li>• Sorting Worksheet</li> </ul> <p>Labs (FJ):</p> <ul style="list-style-type: none"> <li>• TowersOfHanoi (Ex 12.1, pg 462)</li> <li>• Sum</li> <li>• Factorial</li> <li>• Fibonacci</li> <li>• HelloWorld</li> <li>• Sequence1, Sequence2</li> <li>• QuickSort</li> <li>• MergeSort</li> <li>• BinarySort</li> <li>• P12-6</li> <li>• P12-7 – Case Study Many Queens</li> <li>• Grid World Parts 3 &amp; 4</li> <li>• Forum discussing most appropriate sorting algorithms for various scenarios</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapter 12 and do FJC chapters 12 exercises.</li> <li>• Cover FJ chapter 12 and sample program code of a recursive method, Merge Sort, and QuickSort.</li> <li>• Cover how to do complexity analysis for two simple algorithms.</li> <li>• Assign a number of programs using recursion. Have students work with the two new sorts.</li> <li>• Students will be given a worksheet where they must trace through a MergeSort and a QuickSort.</li> <li>• Stress to students how a recursive method calls itself and have students trace a recursive method.</li> <li>• Unit Test #7: written and program</li> </ul>
<p><b>8</b> (25-27)</p>	<p><b>More on Classes, Inheritance, Interfaces, and Polymorphism</b></p> <p><b>Topics: [C5] [C6]</b></p> <ul style="list-style-type: none"> <li>• Static Variables and Methods</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapter 10</li> <li>• BA: Chapter 3</li> </ul>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<ul style="list-style-type: none"> <li>• Turtle Graphics</li> <li>• Java Interfaces</li> <li>• Code Reuse Through Inheritance</li> <li>• Abstract Classes</li> <li>• Polymorphism</li> <li>• Object Equality</li> <li>• javadoc</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Declare and implement static variables and methods in a class.</li> <li>• Understand the role of interfaces and how to define one for a set of classes.</li> <li>• Understand inheritance by extending a class.</li> <li>• Understand the use of polymorphism and how to override methods in a superclass.</li> <li>• Create an abstract class containing common variables and methods.</li> <li>• Understand the equality of reference types.</li> <li>• Know how to create methods that have preconditions and postconditions.</li> <li>• Create javadoc documentation using @param, @return and other common tags</li> </ul>	<p><b>Assignments:</b></p> <p>Written:</p> <ul style="list-style-type: none"> <li>• Labs (FJ):</li> <li>• DrawSquare (Ex 10.1, pg 357)</li> <li>• TestPens (Ex10.2, pg 360)</li> <li>• TestShapes (Ex 10.3, Ex 10.4, pg 362-369, P10-3, P10-4, pg 403)</li> <li>• PetParade (2004 AP Exam Question 2)</li> <li>• Class Hierarchy Forum where students design a class hierarchy related to a vocation using UML</li> <li>• Worksheets on Inheritance, Superclass/Subclass, and Interfaces</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students read FJ chapter 10.</li> <li>• PowerPoint over FJ chapter 10 and go over the Shapes abstract class. .</li> <li>• Show students the three types of relationships among classes: Dependency, Aggregation (has-a), and Inheritance (is-a). Show relationships diagram on page 380 (Figure 10-7).</li> <li>• Unit Test #8: written</li> </ul>
<p><b>9</b> (28-32)</p>	<p><b>GridWorld (Parts 1-4)</b> <b>Topics:</b> [C4] [C5] [C6] [C7]</p> <ul style="list-style-type: none"> <li>• Experiment With a Large Program</li> <li>• Using and Modifying Classes and Methods</li> <li>• Inheritance</li> <li>• Extending a Class</li> <li>• Interfaces</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Run the demo and experiment with different methods to see how they work.</li> <li>• Understand how a large program works by reading the documentation that comes with the case study.</li> <li>• Create and extend a class</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• GridWorld: Parts 1-4</li> <li>• BA: Chapter 13</li> </ul> <p><b>Assignments:</b></p> <p>Written:</p> <ul style="list-style-type: none"> <li>• GW: Written Questions: Sets 1-9</li> <li>• GW Student Exercises and Labs</li> <li>• GW Test</li> <li>• GW Free Response Assignments</li> <li>• Blue Pelican Java Exercises and Labs</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Students need to read the GW case study and work through the</li> </ul>

<b>Unit (Weeks)</b>	<b>Title, Topics, and Student Objectives</b>	<b>Resources, Assignments, and Strategies</b>
	<ul style="list-style-type: none"> <li>• Implement an existing interface</li> <li>• Understand polymorphism and override methods.</li> <li>• Understand how different objects interact with each other.</li> <li>• Make new types of Critters by extending the Critter class.</li> </ul>	<p>exercises from the Blue Pelican Java GridWorld resources (includes videos, exercises, and labs)</p> <ul style="list-style-type: none"> <li>• Each student needs to make the modifications to the case study at their own computer for each of the computer exercises.</li> <li>• Students create different kinds of Critters not mentioned in the documentation.</li> <li>• Unit Test #9: multiple choice and free response</li> </ul>
<p><b>10</b> (33-36)</p>	<p><b>Review for AP Exam</b></p> <p><b>Topics:</b></p> <ul style="list-style-type: none"> <li>• Review For AP Exam</li> <li>• Free-Response Questions</li> <li>• Multiple Choice Questions</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Prepare for the AP Exam by studying practice exams.</li> <li>• Complete Barron’s sample exams.</li> <li>• Study sample Multiple Choice questions.</li> <li>• Discuss scoring of Multiple Choice questions and testing strategy when it comes to guessing at answers.</li> <li>• Study Free-Response Questions</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• Free-response questions from 2004-2007</li> <li>• BA: Practice Exams</li> </ul> <p><b>Assignments:</b></p> <p>Written:</p> <ul style="list-style-type: none"> <li>• Practice Exams: 2004-2007</li> <li>• Barrons Practice Exams</li> <li>• Free Response Test</li> <li>• Final Exam - Multiple Choice Test modeled after AP Exam Multiple Choice Questions</li> </ul> <p><b>Strategies:</b></p> <ul style="list-style-type: none"> <li>• Have students complete Free Response Questions with a timer. Have each student grade their own according to the published rubric.</li> </ul>
<p><b>11</b> (optional 37-39)</p>	<p><b>Creating GUI’s</b></p> <p><b>Topics: [C3] [C4] [C5]</b></p> <ul style="list-style-type: none"> <li>• Windows and Panels</li> <li>• Drawing Shapes and Text</li> <li>• I/O Dialog Boxes</li> <li>• Images and Mouse Events</li> <li>• Timers and Animations</li> <li>• The Model/View/Controller Pattern</li> <li>• HTML and Applets</li> <li>• Student Class GUI</li> <li>• Drawing Multiple Shapes</li> </ul>	<p><b>Resources:</b></p> <ul style="list-style-type: none"> <li>• FJ: Chapters 2-13, 15</li> </ul> <p><b>Assignments:</b></p> <p><b>Labs (FJ):</b></p> <ul style="list-style-type: none"> <li>• Selected Examples and Assignments from the end of chapter sections in Chapters 2-13 and Chapter 15</li> </ul> <p><b>Strategies:</b></p>

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assignments, and Strategies
	<ul style="list-style-type: none"> <li>• Menus</li> <li>• Sliders</li> <li>• Text Areas and Scrolling List Boxes</li> <li>• Check Boxes and Radio Buttons</li> </ul> <p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Write a simple graphics program.</li> <li>• Work with color and text properties.</li> <li>• Create I/O dialog boxes.</li> <li>• Create GUIs with images and mouse events.</li> <li>• Using timers and animations with GUIs.</li> <li>• Create GUIs using the model/view/controller pattern</li> <li>• Convert a Java application to an applet and embed the applet in a Web page.</li> <li>• Changing the view of the student test scores.</li> <li>• Drawing multiple shapes on GUIs.</li> <li>• Add menus to a GUI.</li> <li>• Add sliders to a GUI.</li> <li>• Add text areas and scrolling list boxes to a GUI.</li> <li>• Add check boxes and radio buttons to a GUI.</li> </ul>	<ul style="list-style-type: none"> <li>• Since this material is not needed for the AP exam, it will be covered after the exam is completed.</li> <li>• It will also be used as supplemental sponge assignments throughout the semester for those students who finish work early.</li> </ul>

### AP Curricular Requirements

C2 - The course includes all of the topics listed in the “Computer Science A” column of the Topic Outline in the AP Computer Science Course Description.

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 standard Java library classes from the AP java subset delineated in Appendices A and B of the AP Computer Science Course

Description. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP java subset.)

C8 - 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.

C9 – The course teaches students to recognize the ethical and social implications of computer use.

**Student Evaluation and Grading Policies for Credit Courses Only:**

a. CTD Grading scale

<b>A+</b> 97-100	<b>B+</b> 87-89	<b>C+</b> 77-79	<b>D+</b> 67-69	<b>F</b> Below 60
<b>A</b> 93-96	<b>B</b> 83-86	<b>C</b> 73-76	<b>D</b> 63-66	
<b>A-</b> 90-92	<b>B-</b> 80-82	<b>C-</b> 70-72	<b>D-</b> 60-62	