AP® Computer Science A
Syllabus 2

Course Overview
AP® Computer Science A is both a college-prep course for potential computer science majors and a foundation course for students planning to study in other technical fields such as engineering, physics, chemistry, and geology. The course emphasizes programming methodology, procedural abstraction, and in-depth study of algorithms, data structures, and data abstractions, as well as a detailed examination of a large case study program. Instruction includes preparation for the AP Computer Science A Exam. In teaching this course, my reward comes when students can apply the programming tools they have learned to real-life examples on their own. Computer science is more than just programming. Students should leave my class with a clear understanding of Java and the ability to adapt to any new programming language that they are taught in college. I want them to have the confidence to tackle any problem-solving obstacles they encounter.

Major Texts
Bergin, Joseph et al. *Karel J Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java*. Redwood City, Calif.: Dreamsongs Press, 2005.
http://csis.pace.edu/~bergin/KarelJava2ed/Karel%2B%2BJavaEdition.html

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

Horstmann, Cay. *Big Java*. Hoboken, N.J.: Wiley, 2002.

Lambert, Ken, and Martin Osborne. *Fundamentals of Java, Comprehensive Course.* 2nd ed. Boston: Course Technology, 2002.

Course Planner [C2]

The resources list includes the following text references: *Karel J. Robot* (KJR), *GridWorld Case Study* (MBS), *Big Java* (BJ), and *Fundamentals of Java* (FJ).

Unit (Weeks)	Title, Topics, and Student Objectives	Resources, Assessments, and Strategies
1 (0–3)	**Karel J. Robot** *(Introduces objects and inheritance)* **Topics:** [C3] • Objects • Classes • Looping • Conditionals **Objectives:** • Write and use simple classes with Karel J Robot • Learn the basics of conditionals and looping	**Resource:** 2 KJR **Assessments:** • Program-specific tasks for Karel • Create a SmartRobot Class to teach Karel more commands: turnRight(), turnAround(), climbStair() • Clear a field of beepers (using loops). • Redistribute a field of beepers (using loops and conditionals) • Run a hurdle race: o same height and equally spaced; o same height and unequally spaced; o different heights and unequally spaced.
2 (4)	**Java Basics** **Topics:** [C3] [C8] [C9] • Computer basics • Java basics • Using the compiler • Input and output **Objectives:** • Understand terminology: CPU, system and application software, primary and secondary memory, LAN, WAN, hard disk, CD-ROM • Understand computer ethics such as acceptable use policies, copyright, intellectual property, freeware, shareware, downloading music • Understand how all the different parts of the computer work together • Understand terminology: compiler, IDE, JVM • Edit, compile, and run a simple program in Java • Understand the different compile time errors, runtime errors, and logic errors • Use BufferedReader for input • Use output with System.out using print and println and format output to look nice	**Resource:** • FJ: lesson 3, Critical thinking • FJ: Project 1-1, Critical thinking **Assessments**: • Labs: Triangle, rectangle, square: area, and perimeter program • Get input for the registrar's office program • Label the parts of the computer **Strategies**: • To discuss computer ethics, begin by looking at the school's acceptable use policy, then go to the Web and look at the ACM's code of ethics. Students will write a small paper in favor of or against something related to computer ethics such as making copies of a copyrighted program and giving it away for free. • Assign a lot of small programs that illustrate different types of input and output—make sure students have used every type of input and displayed it in different ways.

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.

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.

3 (5)	**Defining Variables, Arithmetic Expressions** **Topics:** **[C3] [C4] [C5] [C6]** • Using and understanding variables • Comments • Arithmetic expressions in Java programs • Representing numbers in different bases **Objectives:** • Understand terminology: comments, variables, constants, reserved words, literals • Declare and initialize variables and constants in Java • Understand mathematical expressions in Java and their precedence • Understand how to change bases of numbers • Use casting to make their data more accurate • Understand limitations of finite representations of numbers such as the range of integers, real and float • Use the assignment operator correctly	**Resource:** • FJ: lesson 3, Projects **Assessments:** Labs: • Paycheck program; have employee information entered and calculate pay • Modify the paycheck program to also include any overtime hours in the calculations **Strategies:** • Students need practice with how the different types, double and int, relate when they are used in mathematical operations • Present a lot of small program examples in which they have to find the errors
4 (6–7)	**Introduction to Classes and OOP** **Topics:** **[C4] [C5] [C6]** • Creating and using classes **Objective:** • Understand terminology: constructor, accessor, mutator, instance variable, encapsulation, information hiding, procedural abstraction • Understand the difference between public and private access in a class • Use and comprehend the DecimalFormat class and the Random class • Write classes from scratch, choosing appropriate data representation • Understand how to declare a method and declare parameters in that method • Understand the use of preconditions, postconditions and assertions when designing methods	**Resource:** • BJ: chapter 3 **Assessments:** Labs: Purse class and StampMachine class **Strategies:** • Give students classes to complete, in which they are given a description and they must choose appropriate representation for that class

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.)

	• Understand the difference between OOP development and top-down development	
5 (8–12)	**Conditionals and Looping** **Topics:** • if, if-else, while, for **Objectives:** **[C3] [C6]** • Understand terminology: control statements, counter, infinite loop, iteration, nested loops, logical operators, truth tables • Construct syntactically correct loops and conditional statements • Understand the different errors that may occur with loops and employ helpful debugging techniques such as hand-tracing and extra print statements to figure out errors • Use logical operators to make programs more robust • Construct truth tables • Be able to calculate statement execution counts, e.g., how many times did the loop execute?	**Resources:** • FJ: lessons 4 and 6, Projects **Assessments:** Labs: • Approximate PI using Leibniz's method • Base Conversion: Convert from base 10 to base 2 • Guess My Number game • Euclidean algorithm program • Perimeter and area of rectangles using all combinations of certain range **Strategies:** • Students need practice writing different types of loops and conditionals
6 (13–14)	**The String Class** **Topic:** **[C6]** • String class **Objectives:** • Instantiate String objects • Understand that Strings are immutable • Use appropriate String methods to solve problems	**Resource:** • FJ: lesson 10.1 **Assessments:** • FJ: exercise 10.1 • Lab: LineEditor Class (*AP CS Course Description*) **Strategies**: • Work several examples using the substring method
7 (15–17)	**ArrayList** **Topic:** **[C6]** • Using ArrayList class **Objective:** • Use the ArrayList methods	**Resources:** • FJ: lesson 10.7 • BJ: 13.1 and 13.2 **Assessments:** • BJ: exercise p.13.1 • WordList (2004 AP CS A Exam, Free-Response Question 1, AP Central®) **Strategies:** • Stress the difference between add and set • Draw pictures of the ArrayList after add, set, and remove have been performed

C3—The course teaches students to design and implement computer-based solutions to problems in a variety of application areas.

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.)

8 (18)	**Arrays** **Topics:** **[C4] [C5] [C6]** • Declaring and initializing arrays • Manipulating arrays with loops • Creating parallel arrays **Objectives:** • Understand terminology: array, element, index, logical size, physical size, parallel arrays • Declare one-dimensional arrays in Java • Use initializer lists when declaring arrays • Manipulate arrays using loops and array indices • Use the physical and logical size of an array together to guarantee they do not go beyond the bounds of their array by identifying the boundary cases and using test data to verify results • Understand how parallel arrays can be useful when processing certain types of data • Work with arrays of primitive data types as well as arrays of objects while understanding the difference between the two types of data • Understand when to choose an array to represent data instead of an ArrayList	**Resource:** • FJ: lesson 8, Projects **Assessments:** Lab: • For one-dimensional arrays, read in numbers and place each one in an even, odd, and/or negative list **Strategies:** • Students need practice manipulating loops that work with arrays • Students also need to be reminded about the indexing of arrays beginning at zero
9 (19–21)	**Searching and Sorting Arrays** **Topics:** • Bubble, Selection, Insertion sorts • Sequential and Binary searches **Objectives:** **[C4] [C5] [C6]** • Write a method for searching an array • Perform insertions and deletions at given positions in arrays • Trace through sorting and searching algorithms and understand time constraints of each • Understand the algorithms behind each of the following searching and sorting techniques: bubble, selection, and insertion sorts; sequential search and binary search • Understand the time efficiency	**Resource:** • FJ: lesson 10 **Assessments:** Lab: • Students make their own "utility" class that includes all of these sorts and searches **Strategies:** • Students need practice tracing through sorts and searches and determining the runtime of each • Students also do well with a worksheet that addresses the efficiency of each of the strategies they have learned, efficiency for a sorted versus unsorted list, and "best," "worst," and "average" efficiency

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.)

	of each sort and search and when it is desirable to use each one • Identify reusable components from existing code using classes and class libraries • Given different scenarios, students should be able to choose the most appropriate sort or search	
10 (22–24)	**GridWorld (Parts 1–3)** **Topics:** **[C6] [C7]** • Experimenting with a large program • Using classes • Modifying classes **Objectives:** • Run the case study and analyze output • Understand how the development of a large program came about by reading the chapters of the case study • Observe and experiment with the GridWorld case study • Understand the Bug class, Runner class, Grid Interface • Extend the Bug class by creating a specialized bug to meet some new type of bug requirement	**Resource:** • GridWorld: Parts 1–3 **Assessments:** • Exercises from within the case study **Strategies**: • Read the manual for the case study thoroughly • Be familiar with all the classes and interfaces discussed
11 (25–27)	**More on Classes, Inheritance, Interfaces** **Topics:** **[C5] [C6]** • Classes • Inheritance • Abstract classes • Interfaces **Objectives:** • Demonstrate inheritance by extending a class • Understand polymorphism and know when it is appropriate to override methods in a super class • Create and extend an abstract class • Create and extend a class given class specifications with the relationships among the classes described • Implement an interface	**Resources:** • BJ: chapter 11 • FJ: lessons 9.5 & 9.6 **Assessments:** • Create an abstract Shape class. • Pet Parade (2004 AP CS A Exam: Free-Response Question 2, on AP Central) **Strategies**: • Draw pictures of the inheritance hierarchy • Note: This unit could be moved to after unit 12 to use the GridWorld Case Study to introduce inheritance

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.)

C7—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®.

12 (28–29)	**GridWorld (Part 4)** **Topic:** **[C4] [C5] [C6]** • Inheritance **Objective**: • Use inheritance to extend the Critter Class by making new types of Critters	**Resource:** • MBS: chapter 4 **Assessments:** • Exercises from the text **Strategies**: • Have fun with this chapter • Allow the students to be creative after working through the exercises and analysis • Create different kinds Critters
13 (30–31)	**Recursion (and Merge Sort)** **Topics:** **[C4] [C5] [C6]** • Recursion • Merge Sort **Objectives:** • Create a recursive method to solve a problem • Understand the difference between recursive and iterative solutions to a problem • Understand and use the Merge Sort • Understand how to calculate the informal runtime of merge sort and compare it's running time to the other sorts already learned	**Resources:** • FJ: lesson 11.1 • BJ: section 18.4 **Assessments:** • Factorial program • Rewrite loop programs with recursion **Strategies**: • Ask, "What is returned by this method?"
14 (32–36)	**Review** **Topics:** • Review AP Computer Science A topics. **Objective:** • Prepare for the AP CS A Exam by reviewing material and taking practice exams	**Resources:** • Previous free-response questions from AP Central **Assessments:** • Practice exams

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.)

Teaching Strategies

I strive to create a learning environment that is comfortable for all students. Those who have never touched a computer should be as at ease in my class as those who have taught themselves how to program. I aim to foster critical thinking, a lifelong skill, and I accomplish this by giving challenging, yet not impossible, assignments. When new topics are introduced, I use a hands-on approach of having students see and run examples. While the novices ask questions, more experienced students can make changes to the examples and experiment with different outcomes.

Experienced programmers help the novices in a mentoring program after school. This promotes student leadership and propels in-class learning.

Lab Component

I give at least two programs per unit, and students work on programs about 70 percent of the time. They can also come in before or after school for extra

programming time and help. All computers have *JCreator LE* installed, and students have access to information on how to download it at home.