Curriculum Map

Grade(s): 11-12   Subject: AP Computer Science

(Semester 1 - Weeks 1-18)				
Unit / Weeks	**Content**	**Skills**	**Assessments**	**Standards**
Lesson 1 - Background (Weeks 1-3) - *1.1 History of Computers* - *1.2 Computer Hardware and Software* - *1.3 Binary Representation and Information and Computer Memory* - *1.4 Programming Languages* - *1.5 The Software Development Process* - *1.6 Basic Concepts of Object-Oriented Programming*	Chapter 1 of Textbook *(Fundamentals of Java)* This is the only chapter in the book that is not about the details of writing Java programs. It discusses computing in general, hardware and software, the representation of information in binary (i.e., as 0s and 1s), and general concepts of object-oriented programming. All material is structured to give a broad understanding of computing and a foundation for the study of programming. -	To give a brief history of computers. To describe how hardware and software make up computer architecture. To understand the binary representation of data and programs in computers. To discuss the evolution of programming languages. To describe the software development process. To discuss the fundamental concepts of object-oriented programming.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
Lesson 2 – Fist Java Programs (Weeks 3-6) - *2.1 Why Java?* - *2.2 The Java Virtual Machine and Byte Code* - *2.3 Choosing a User Interface Style* - *2.4 Hello World* - *2.5 Edit, Compile, and Execute*	Chapter 2 of Textbook *(Fundamentals of Java)* Programs are written in programming languages, and the language used in this book is Java. This lesson has the student begin programming with a couple of simple Java programs. They are shown how to write these first programs, compile them, and run them. In the process, they become	To discuss why Java is an important programming language. To explain the Java virtual machine and byte code. To choose a user interface style. To describe the structure of a simple Java program. To write a simple program. To edit, compile, and run a	Website Analysis Activity Quizzes from Website – Java, comOOP. Completion of 1st Program(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 2 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)

- *2.6 Temperature Conversion* - *2.7 Turtle Graphic*	acquainted with a Java programming environment, the structure of a simple Java program, and the basic ideas of variables, input and output statements, and sending messages to objects. -	program using a Java development environment. To format a program to give a pleasing, consistent appearance. To understand compile-time errors. To write a simple turtle graphics program.		
Lesson 3 – Syntax Errors, and Debugging (Weeks 6-9) - *3.1 Language Elements* - *3.2 Basic Java Syntax and Semantics* - *3.3 Terminal I/O for Different Data Types* - *3.4 Comments* - *3.5 Programming Errors* - *3.6 Debugging* - *3.7 Graphics and GUIs: Drawing Shapes and Text*	Chapter 3 of Textbook *(Fundamentals of Java)* Students are shown how to use a programming language, become familiar with its vocabulary and the rules for forming grammatically correct statements. The student also learns how to construct meaningful statements and statements that express the programmer's intent. Errors of form, meaning, and intent are possible, so finally one must know how to detect these errors and correct them. This lesson discusses the basic elements of the Java language in details and explores how to find and correct errors in programs. Graphics are included!	To construct and use numeric and string literals. To name and use variables and constants. To create arithmetic expressions. To understand the precedence of different arithmetic operators. To concatenate two strings or a number and a string. To know how and when to use comments in a program. To tell the difference between syntax errors, run-time errors, and logic errors. To insert output statements to debug a program.	Website Analysis Activity Quizzes from Website – Basic Elements, Errors, Simple Rules and . Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 3 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer) End of 1st Quarter Final Exam (Chapters 1-3)	College Board - AP Computer Science (http://www.collegeboard.com/)

Lesson 4 – Control Statements (Weeks 9-12) - *4.1 Additional Operators* - *4.2 Standard Classes and Methods* - *4.3 A Shortcut for Inputting Data* - *4.4 A Visit to the Farm* - *4.5 The* ***if*** *and* ***if-else*** *Statements* - *4.6 The* ***while*** *Statement* - *4.7 The* ***for*** *Statement* - *4.8 Nested Control Statements and the* ***break*** *Statement* - *4.9 Errors in Loops*	Chapter 4 of Textbook *(Fundamentals of Java)* All the programs to this point have consisted of short sequences of instructions that are executed one after the other. The student is presented with the concept that such a scheme, even if we allowed the sequence of instructions to become extremely long, would not be very useful. The Students examine how computer programs, as in real life, instructions must express repetition and selection. Expressing these notions in Java is the major topic of this lesson, but before doing so we present a couple of topics that we will use throughout the rest of the lesson.	To use the increment and decrement operators. To use standard math methods. To use if and if-else statements to make choices. To use while and for loops to repeat a process. To construct appropriate conditions for control statements using relational operators. To detect and correct common errors involving loops.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 4 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
Case Study: Grid World Part 1 (Weeks 12-13) - *Pages 1-4: Running the Demo* - *Pages 5-7: Exploring Actor State and Behavior*	AP® Computer Science in Java ~ GridWorld Case Study Students are introduced to the AP Case Study that will be included in at least 2 of the 4 open-ended questions on the AP A exam.	To load and run the Case Study environment, examining the basic attributes of the Java GridWorld environment.	Observation Checklist Rubric Class discussion; GridWorld Quiz 1 (25 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)

Lesson 5 – Defining Classes Intro (Weeks 13-15) - *5.1 The Internal Structure of Classes and Objects* - *5.2 A Student Class* - *5.3 Editing, Compiling, and Testing the Student Class* - *5.4 The Structure and Behavior of Methods* - *5.5 Scope and Lifetime of Variables* - *5.6 Turtle Graphics: Colors, Pen Widths, and Movement*	Chapter 5 of Textbook *(Fundamentals of Java)* We introduced basic object-oriented terminology in Lesson 1 and have used it repeatedly since then. Until now, we have focused on choosing among predefined classes to solve problems. We have shown how to declare variables of different classes, assign objects to these variables, and send them messages. In this lesson, we explore the internal workings of objects. We introduce the basic structure of class definitions, so that you will be able to read and modify classes and create classes of your own. We restrict our focus to a few simple concepts and add more details in later lessons.	To design and implement a simple class from user requirements. To organize a program in terms of a view class and a model class To use visibility modifiers to make methods visible to clients and restrict access to data within a class. To write appropriate mutator methods, accessor methods, and constructors for a class. To understand how parameters transmit data to methods. To use instance variables, local variables, and parameters appropriately. To organize a complex task in terms of helper methods.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 5 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
Lesson 6 – Control Statements Continued (Weeks 15-18) - *6.1 Logical Operators* - *6.2 Testing **if** Statements* - *6.3 Nested **if** Statements* - *6.4 Logical Errors in Nested **if**s* - *6.5 Nested Loops* - *6.6 Testing Loops*	Chapter 6 of Textbook *(Fundamentals of Java)* This lesson explores more advanced aspects of the control statements introduced in Chapter 4. Topics include logical operators, nested **if** statements, and nested loops. The lesson also described strategies for testing programs that contain control statements. Programmers try to write programs that are free of	To construct complex Boolean expressions using the logical operators && (AND), || (OR), and ! (NOT). To construct truth tables for Boolean expressions. To understand the logic of nested if statements and extended if statements. To test if statements in a comprehensive manner.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 6 Exam (50 Questions: Multiple Choice, True/False, Fill	College Board - AP Computer Science (http://www.collegeboard.com/)

	logic errors, but they seldom success. Consequently, they must test their programs thoroughly before releasing them--and even so, errors will still slip through. Notice that we say "will" instead of "might." Software is so incredibly complex that no significant software product has ever been released free of errors; however, the situation would be much worse if we stopped emphasizing the importance of testing.	To construct nested loops. To create appropriate test cases for if statements and loops. To convert text to numbers (Strings to ints and doubles) and numbers to text. To format a decimal number to a desired number of decimal places. To left and right justify text. To left and right justify numerical values. To arrange text and numerical values of various formats in columned tables.	in the Blank, and Short Answer)	

(Semester 2- Weeks 19-33)				
Unit / Weeks	**Content**	**Skills**	**Assessments**	**Standards**
Lesson 7 – Improving the User Interface (Weeks 19-21) - *7.1 A Thermometer Class* - *7.2 Repeating Sets of Inputs and Type* ***char*** - *7.3 A Menu-Driven Conversion Program* - *7.4 A GUI-Based Conversion Program* - *7.5 The GUI Program Explained* - *7.6 Other Window Objects and Methods*	Chapter 7 of Textbook *(Fundamentals of Java)* We do not judge a book by its cover because we are interested in its contents, not its appearance. However, we do judge a software product by its interface because we have no other way to access its functionality. In this lesson, we explore several ways to improve a program's interface. First, we present some standard techniques for enhancing terminal-based interfaces, and then we show how to develop graphical user	To construct a query-driven terminal interface. To construct a menu-driven terminal interface. To construct a graphical user interface. To position window objects in a grid within a window. To write a method to handle users' interactions with command buttons. To manipulate window objects to input and output integers,	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)

- *7.7 Formatted Output* - *7.8 GUIs and Applets*	interfaces (GUIs). The transition to GUIs involves making two adjustments to our thinking. First, the structure of a GUI program differs significantly from that of a terminal-based program. Second, a GUI program is event driven, meaning that it is inactive until the user clicks a button or selects a menu option. In contrast, a terminal-based program maintains constant control over the interactions with the user.	doubles, and text.		
Case Study: Grid World Part 2 (Weeks 21-23) *Part 2* - *Pages 1-3: The BoxBugRunner project* - *Pages 4-5: Exercises 1-4*	AP® Computer Science in Java ~ GridWorld Case Study Exercise 1 introduces the concepts and attributes of the DiamondBug class. Exercise 2 presents an overview of DiamondBug's act() method Exercise 3 adds a new class named CircleBug	Since all future calls of the `DiamondBug`'s `act()` method are inherited from the `BoxBug` which are 90 degree turns, the `DiamondBug` will continue moving in diagonal directions and thus will trace out the path of a diamond rather than a square.	Observation Checklist Rubric Class discussion; GridWorld Quiz 2 (25 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
Lesson 9 – Classes (Weeks 23-26) - *9.1 Class **(static)** Variables and Methods* - *9.2 Java Interfaces--The Client Perspective* - *9.3 Java Interfaces--The Implementation Perspective* - *9.4 Code Reuse*	Chapter 9 of Textbook *(Fundamentals of Java)* Lesson 5 presented an overview of classes that allowed you to learn to read, modify, and define them. The examples of classes discussed there are simplified versions of what real programmers would see, however. Defining classes is only one aspect of object-oriented	To know when it is appropriate to include class (static) variables and methods in a class. To understand the role of Java interfaces in a software system and define an interface for a set of implementing classes.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric	College Board - AP Computer Science (http://www.collegeboard.com/)

Through Inheritance - *9.5 Inheritance and Abstract Classes* - *9.6 Some Observations About Interfaces and Inheritance* - *9.7 Acceptable Classes for Parameters and Return Values* - *9.8 Error Handling with Classes* - *9.9 Exceptions* - *9.10 Reference Types, Equality, and Object Identity*	programming. The real power of object-oriented programming comes from its capacity to reduce code and to distribute responsibilities for such things as error handling in a software system. This capability can be exploited only when you have an understanding of some related concepts. Some of these concepts (see pages 269-270 in text) are: - Static Variables and Methods - Interfaces - Inheritance - Abstract Classes - Polymorphism - Preconditions and Postconditions - Exceptions for Error Handling - Reference Types	To understand the use of inheritance by extending a class. To understand the use of polymorphism and know how to override methods in a superclass. To place the common features (variables and methods) of a set of classes in an abstract class. To understand the implications of reference types for equality, copying, and mixed-mode operations.	Class discussions and observations Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	
Lesson 10 – Arrays Continued **(Weeks 26-28)** - *10.1 Advanced Operations on Strings* - *10.2 Searching* - *10.3 Sorting* - *10.4 Insertions and Removals* - *10.5 Working with*	Chapter 10 of Textbook *(Fundamentals of Java)* Earlier in the course, we examined how to declare array variables, instantiate array objects, and manipulate arrays using the subscript operator, loops, and methods. The current lesson covers more complex operation on arrays, such as searching sorting, insertions, and removals.	Use string methods appropriately. Write a method for searching an array. Understand why a sorted array can be searched more efficiently than an unsorted array.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations	College Board - AP Computer Science (http://www.collegeboard.com/)

Arrays of Objects - *10.6 Swing GUI: Menus, Scrolling List Boxes, and Dialogs* - *10.7 The Class* ***java.util.ArrayList***	Along the way, we examine issues that arise in the use of arrays of objects and we introduce Java's ArrayList class. As a prelude to these topics, we begin with a look at some advanced string methods...	Write a method to sort an array. Write methods to perform insertions and removals at given positions in an array. Understand the issues involved when working with arrays of objects. Perform simple operations with Java's **ArrayList** class	Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	
Case Study: Grid World Part 3 (Week 28) - *Pages 5-7: GridWorld Classes and Interfaces*	AP® Computer Science in Java ~ GridWorld Case Study As an integration of the Case Study into the core curriculum of the AP class, the in-class lab activity will be Jumper, in groups of 4, with specification and design due by end of class.	Add a new project named DancingBugRunner to the GridWorld workspace. Then add a new class named DancingBug to the DancingBugRunner project.	Observation Checklist Rubric Class discussion; GridWorld Quiz 3 (25 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
Lesson 11 – Recursion (Weeks 28-30) - *11.1 Advanced Operations on Strings* - *11.2 Searching* - *11.3 Sorting* - *11.4 Insertions and Removals* - *11.5 Working with Arrays of Objects* - *11.6 Swing GUI: Menus, Scrolling List Boxes, and Dialogs* - *11.7 The Class java.util.ArrayList*	Chapter 11 of Textbook *(Fundamentals of Java)* In Lesson 9, we examined how to declare array variables, instantiate array objects, and manipulate arrays using the subscript operator, loops, and methods. The current lesson covers more complex operation on arrays, such as searching sorting, insertions, and removals. Along the way, we examine issues that arise in the use of arrays of objects and we introduce Java's ArrayList class.	ObjectivesUse string methods appropriately. Write a method for searching an array. Understand why a sorted array can be searched more efficiently than an unsorted array. Write methods to perform insertions and removals at given positions in an array. Write a method to sort an array.	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)

	As a prelude to these topics, we begin with a look at some advanced string methods...	Bubble Sort Selection Sort Insertion Sort Understand merge concepts in preparation for working with the recursive Merge Sort in the next chapter. Understand the issues involved when working with arrays of objects. Perform simple operations with Java's templated ArrayList class		
Lesson 12 **(Weeks 30-31)** - *Recursion* - *Complexity Analysis (THIS SECTION Was OMITTED)* - *Binary Search* - *Quicksort*	Chapter 12 of Textbook *(Funda mentals of Java)* In this lesson, formally introduce recursion. The additional topic in the text, complexity analysis, will be omitted from this course because it is part of the B curriculum. A recursive algorithm is one that refers to itself by name in a manner that appears to be circular. Everyday algorithms, such as a recipe to bake cake or instructions to change car oil, are not expressed recursively, but recursive algorithms are common in computer science.	To design and implement a recursive method to solve a problem. To understand the similarities and differences between recursive and iterative solutions of a problem. To check and test a recursive method for correctness. To understand how a computer executes a recursive method. *For the record, these additional text objectives will be omitted:* *To perform a simple complexity analysis of an algorithm using big-O notation.* *To recognize some typical orders of complexity.* *To understand the behavior of a complex sort algorithm such*	Website Analysis Activity Quizzes from Website – History, Hardware, Memory, Languages, Software, OOP. Completion of Site(s)– Rubric assessment Observation Checklist Rubric Class discussions and observations Chapter 1 Exam (50 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)

		as the quicksort.		
Case Study: Grid World Part 4 **(Weeks 31-32)** - *Trey Carroll's Flash Screen-capture Lecture* - *Pages 8-10: Exploring Actor State and Behavior*	AP® Computer Science in Java ~ GridWorld Case Study Trey Carroll's Lecture(s) walk through all the fundamental concepts. Students review course content covered that are specific to the Case Study.	Running the Demo & Exploring Actor State and Behavior - View Videos 1 to 4.	Observation Checklist Rubric Class discussion; GridWorld Quiz 4 (25 Questions: Multiple Choice, True/False, Fill in the Blank, and Short Answer)	College Board - AP Computer Science (http://www.collegeboard.com/)
AP Exam Review, Diagnostic Exams & AP Multiple Choice Practice Exams **(Weeks 32)**				College Board - AP Computer Science (http://www.collegeboard.com/)
Final Grid World Review & AP Open Ended Questions Practice Exams **(Weeks 32-33)**				College Board - AP Computer Science (http://www.collegeboard.com/)
AP Exam Week →				