

Syllabus: AP Computer Science A

Date: January 7, 2011

Overview

Course Objectives

The proposed syllabus is for a three trimester long related course, totaling 15 weeks. Students will work a minimum of 10 hours per week from 8:30 to 11 am each day (with a 15 minute break). Additional time to work on projects will be allotted in the afternoon. The course includes several individual programming projects assigned for one or two weeks each. The learning objectives for this course are as follows:

- Understand and apply the main principles of object-oriented software design and programming: classes and objects, constructors, methods, instance and static variables, inheritance, class hierarchies, and polymorphism
- Learn to code fluently in Java in a well-structured fashion and in good style; learn to pay attention to code clarity and documentation
- Learn to use Java library packages and classes within the scope of the AP Java subset
- Understand the concept of an algorithm; implement algorithms in Java using conditional and iterative control structures and recursion.
- Learn to select appropriate algorithms and data structures to solve a given problem.
- Compare efficiency of alternative solutions to a given problem.
- Learn common searching and sorting algorithms: Sequential Search and Binary Search; Selection Sort, Insertion Sort, and Mergesort
- Understand one- and two-dimensional arrays, the `List` interface, and the `ArrayList` class, and use them appropriately in programming projects
- Acquire skills in designing object-oriented software solutions to problems from various application areas
- Learn the GridWorld case study and accompanying exercises and questions provided by the College Board
- Discuss ethical and social issues related to the use of computers
- Prepare for the AP exam in computer science

Resources

- Lenovo Workstations running Windows 7
- Internet Access
- Software for Java Development from <http://www.oracle.com>
- Java IDE (TBD)
- GridWorld case study at:
http://apcentral.collegeboard.com/apc/public/repository/GridWorld_AP_CompScience.pdf

- Litvin, Maria, and Gary Litvin. *Java Methods: Object-Oriented Programming and Data Structures, 2nd AP Edition*, Andover, Mass.: [Skylight Publishing](#), 2011.
- Litvin, Maria, and Gary Litvin. *Be Prepared for the AP Computer Science Exam in Java, 4th Edition*, Andover, Mass.: [Skylight Publishing](#), 2009.
- *codingbat.com*: <http://codingbat.com/java>
- Litvin, Maria, and Gary Litvin. *250 Multiple-Choice Computer Science Questions in Java*. Andover, Mass.: [Skylight Publishing](#), 2008.
- Current media sources and Internet articles and blogs discussing ethical and social issues related to computer use

Teacher Materials

- The College Board's Computer Science A Course Description
- The College Board's GridWorld Teacher Manual
- *AP Central* resources
- *Java Methods* student files, teacher files, Powerpoints, *Test Package*, additional resources at <http://www.skylit.com/javamethods> and <http://www.skylit.com/oop>

Course Outline

Chapter numbers for readings and exercises refer to *Java Methods, 2nd AP Edition*. The labs, case studies, and projects proposed below come from *Java Methods*. GridWorld refers to the College Board's GridWorld case study narrative. Midterm and trimester exams will be comprised of questions from AP practice exams.

Unit 1: An introduction to computers and software engineering

Week 1: An Introduction to hardware, software and the Internet

- Elements of a computer system
- How information is represented in computer memory
- Binary and hex number systems and ASCII / Unicode
- An introduction to the Internet
- Getting familiar with the software development process
- Compilers and interpreters
- JDK tools (*javac, java, appletviewer, javadoc*)
- Using an IDE
- Java classes and source files
- A brief introduction to OOP

Reading and exercises: Chapter 1

Lab: Find and explore the home pages of some Internet and World Wide Web pioneers

Reading and exercises: Chapter 2

Lab: Compile and run simple programs using command-line JDK tools or an IDE (Section 2.4)

Lab: Compile and run simple GUI applications and an applet (Section 2.6)

Unit 2: Objects, algorithms, and syntax (7 weeks)

Week 2: A first look at objects and classes using the GridWorld context

- Classes and objects
- Classes and source files
- *Case study:* GridWorld
- CRC cards
- Library classes and packages
- The `import` statement
- A first look at fields, constructors, and methods of a class
- Inheritance
- Extending library classes

Reading and exercises: Chapter 3; GridWorld Student Manual Parts 1 and 2

Lab: Set up the first GridWorld project and run BugRunner

Lab: Interacting with “actors” in GridWorld (Section 3.4)

Lab: A new type of actor: RandomBug (Section 3.7)

Lab: Circle and Cylinder classes (Exercise 12, p. 74)

Week 3: Algorithms; Java syntax and style

- The concept of an algorithm
- Pseudocode and flowcharts
- Iterations
- Recursion
- Working with lists
- Case study: Euclid’s GCF Algorithm. (Most of the exercises for this chapter are pencil-and-paper exercises.)
- Syntax and style in a programming language
- Comments
- Reserved words and programmer-defined names
- Statements, braces, blocks, indentation
- Syntax errors, run-time errors, logic errors

Reading and exercises: Chapter 4

Lab: Print stars using iterations and recursion (Exercise 10, p. 102)

Reading and exercises: Chapter 5; Appendix A

Lab: Correcting syntax errors and a logic error as an “adventure game” (Section 5.6)

Midterm Exam: Terminology and Coding examples: Units 1 and 2

Unit 3: Arithmetic, logic, and control statements

Week 4: Data types, variables, and arithmetic

- The concepts of a variable and a data type
- Declarations of variables
- Fields vs. local variables
- The primitive data types: `int`, `double` and `char`
- Literal and symbolic constants
- Initialization of variables
- Scope of variables
- Arithmetic expressions
- Data types in arithmetic expressions
- The cast operator
- The compound assignment (`+=`, etc.) and increment and decrement operators (`++`, `--`)
- Converting numbers and objects into strings

Syllabus based on Syllabus by Skylight Publishing

Found at: <http://www.skylit.com/javamethods/syllabi/index.html>

Reading and exercises: Chapter 6

Lab: Exercises for Chapter 6 (for example, 16, 17, 18, pp. 153-154)

Lab: *Pie Chart* (Section 6.10)

Lab: *Rainbow* (Exercise 19, p. 155)

Week 5: Control Structures

- The `if-else` statement
- Boolean expressions; The `boolean` data type; `true` and `false` values
- Relational and logical operators
- De Morgan's laws
- Short-circuit evaluation
- Nested `if-else` and `if-else-if`
- *Case Study: Craps*. Elements of object-oriented design in *Craps*
- The `switch` statement
- `enum` data types
- `while`, `for`, and `do-while` loops
- `break` and `return` in loops

Reading and exercises: Chapter 7

Lab: Exercises for Chapter 7 (for example, 2, 11, 14-17)

Lab: The `CrapsGame` class for *Craps*: fill in the blanks and test in isolation (Section 7.9)

Lab: Finishing and testing the *Craps* program (Section 7.12)

Extra: `codingbat.com` *Logic-1* and *Logic-2*

Trimester Exam: Terminology and coding examples

Reading and exercises: Chapter 8

Lab: Exercises for Chapter 8 (for example, 1 - 3, p. 212)

Lab: *Perfect Numbers* (Section 8.6)

Unit 4: Classes and class hierarchies

Week 6: Details of defining classes and using objects

- Public and private fields and methods
- Constructors and the `new` operator
- References to objects
- Calling methods and accessing fields
- Passing parameters to constructors and methods
- `return` statement
- Overloaded methods
- Static variables and methods

Reading and exercises: Chapter 9

Lab: Snack Bar

Lab: GridWorld exercises for Part 2 (page 12)

Lab: Snack Bar Continued

Exam: Trimester exam, Units 1-3

Week 7: Strings

- String objects
- Literal strings
- Immutability
- String methods
- Converting strings into numbers and numbers into strings
- The Character class and its methods

Reading and exercises: Chapter 10

Lab: Lipograms (Section 10.8)

Extra: codingbat.com String-1, String-2, String-3

Weeks 8 and 9: Class hierarchies, abstract classes, and interfaces

- Class hierarchies
- Abstract classes
- Invoking superclass's constructors and calling superclass's methods
- Polymorphism
- Interfaces

Reading and exercises: Chapter 11, GridWorld Part 3.

Lab: A GridWorld Dance (Section 11.6)

Lab: Past AP free-response questions on class hierarchies and polymorphism

Lab: Creating a subclass of Actor, GridWorld Part 3 group activity (GridWorld Student Manual p. 24)

Exam: Classes

Unit 5: Arrays, the List interface, the ArrayList class, searching and sorting

Week 10: One- and Two-Dimensional Arrays

- One-dimensional arrays
- Arrays as objects
- Declaring and initializing
- Indices
- Length
- `IndexOutOfBoundsException`
- Two-dimensional arrays
- Accessing the number of rows and columns
- Traversals and the “for-each” loop
- Inserting and removing elements

Reading and exercises: Chapter 12

Lab: *Fortune Teller* (Section 12.3)

Lab: Past free-response questions on arrays

Lab: *Chomp* (Section 12.5)

Extra: codingbat *Arrays-1, Arrays-2, Arrays-3*

Week 11: ArrayLists and GridWorld Critters

- ArrayList’s structure
- The List interface
- ArrayList’s constructors and methods
- Pitfalls
- ArrayLists in GridWorld

Reading and exercises: Chapter 13, GridWorld Student manual Part 4

Lab: *Creating an Index for a Document* (Section 13.5)

Lab: Past AP free-response questions on ArrayList

Lab: GridWorld Part 4 exercises (GridWorld Student Manual, p. 32)

Lab: GridWorld Critters (Section 13.6)

Week 12: Searching and sorting. Introduction to analysis of algorithms

- Comparing objects
- The `equals` method and the `Comparable` interface
- Sequential and Binary Search
- Selection Sort, Insertion Sort, and Mergesort
- The `java.util.Random` class
- The number of comparisons required in Sequential and Binary Search
- Comparison of efficiency of “quadratic” sorting algorithms (Selection Sort and Insertion Sort) vs. Mergesort

Reading and exercises: Chapter 14

Lab: Chapter 14 exercises (for example, 4, 9 pp. 408-409)

Lab: *Keeping Things in order* (Section 14.4)

Lab: *Benchmarks* (Section 14.9) — compares efficiency of several sorting algorithms

Exam: *Units 4-6*

Unit 6: Review

Week 13: GridWorld review

- Review of the GridWorld classes and interfaces
- Modifications and exercises

Reading and exercises: GridWorld Parts 1-4; *Be Prepared* Chapter 6

Labs: *GridWorld Enhancements* (from suggested exercises for Part 4, p. 32, and *Be Prepared*)

Weeks 14/15: Review and practice for the AP exam

Reading:

Be Prepared Chapters 1-5

Be Prepared Chapter 7 (past free-response questions and solutions)

Be Prepared practice exams, *250 Multiple-Choice Computer Science Questions*