Homeschool Programming, Inc.

TeenCoder: Java Programming

Online Course Syllabus and Planner

Updated November, 2015

Online Course Overview

Course Title: *TeenCoder: Java Programming*

Online ISBN: **978-0-9887070-2-3**, published 2015 by CompuScholar, Inc.

Length: 2 semesters

Student Pre-Requisites: Basic computer literacy skills, 9th – 12th grade status.

Description: The *TeenCoder: Java Programming* curriculum is a one-year (two-semester) course covering introductory Java language skills and all topics found on the **AP "Computer Science A"** exam. Information on using this course as preparation for the AP "Computer Science A" exam (including a College Board-approved syllabus) can be found on our course description page:

http://www.homeschoolprogramming.com/teencoder/teencoder_jv_series.php

Other introductory programming courses are not required; students merely need to have typical computer usage skills prior to starting this course.

Materials:

- Online Student and Teacher logins
- Oracle JDK, Eclipse IDE (or alternate IDE at teacher's discretion)
- Windows or Mac OS personal computer
- Course activities (hands-on programming assignments)
- Course instructional videos, lesson text, and supplemental documentation

Labs and Grading

Every chapter contains one or more hands-on programming labs where students will create programs to demonstrate understanding of the lesson topics. Projects begin simply with console Java programs and expand into graphical Java programs later in the course. These labs, combined with individual lesson quizzes and end-of-chapter tests, are used to determine the student grade.

Please Note: Some course lessons and activities may include opportunities for students to work in teams or small groups. If your student is working individually (e.g. as a homeschooler), simply complete those activities as an individual. One student can perform each of the tasks that would be spread across multiple team members in a group setting. Other lesson sections with peer-supported vocabulary re-enforcement or feedback tasks can be skipped or completed with a teacher, if desired.

Course Planner

A typical school year consists of approximately 36 calendar weeks or 180 days of school. This course plan covers 30 school weeks of core and AP exam-prep material from late August through mid-April, leaving time prior to the AP exam for review, practice, and make-up work. The plan assumes students are working 3-5 hours per week to stay on pace. Some students may move faster or slower than the suggested pace.

Each chapter contains multiple lesson quizzes and a chapter test in addition to the listed Lab assignments. Teachers may choose to add Supplemental Lessons as desired to meet student interest. AP-centric material can be ignored for students not taking the AP exam.

Week	Reading and Objectives	Labs
1	Chapter One: Understanding Computer Programming • A Survey of Computer Hardware • Introduction to Computer Software • Common Programming Languages • Computer Ethics and Security	**Establish Development Environment** - Install JDK, create working directory, practice submitting projects through the online interface. Class discussion and review of a sample EULA terms and conditions.

Week	Reading and Objectives	Labs
2	Chapter Two: Getting Started with Java • The Java Platform • Writing Your First Program • Building and Running from the Command Line • Java Classes and Packages	**Show Time!** – The student's first Java program will print the current time to the console. The student will compile and run the program from the command line (without an IDE).
3	Chapter Three: The Eclipse IDE • Introducing Eclipse • Eclipse Java IDE Walk-through • Creating an Eclipse Project • Help and Reference Documentation	**Install Eclipse IDE** – If not already installed, the student will add the Eclipse IDE to their home or school computer. **Eclipse Show Time Project** – The student will recreate the same Show Time project using the Eclipse IDE to write, build, and run the program.
4	Chapter Four: Data Types and Variables • Primitive Data Types • Variables • Printing Data	**Experiment with Data Types** – The student will demonstrate declaring, initializing, and printing variables of different data types.
5	Chapter Five: Working With Strings • Reference Data Types • Comparing Strings • Common String Operations • Formatting and Building Strings • Converting Between Strings and Numbers	**String Theory** – The student will create multiple strings and perform a variety of operations on them, including comparison, substrings, formatting, parsing, and case conversion.

Week	Reading and Objectives	Labs
6	Chapter Six: User Input • Using Command-Line Parameters • Interactive User Input • Validating User Input	**Conversation Piece** – The student will create a program using a command-line Scanner to obtain a variety of user input, and then format that input into an output story.
7	Chapter Seven: Basic Flow Control • Logical Expressions and Relational Operators • Using the "if" Statement • The "switch" Statement • For Loops • While Loops	**Fun Factorials** – The student will demonstrate use of a for() loop, while() loop, and do-while() loop to calculate factorials of an input number. Boundary conditions involving maximum integer sizes are explored and tested.
8	Chapter Eight: Writing Methods • Writing and Calling Methods • Method Parameters and Return Values • Calling Methods	**Checkerboard** – The student will write a program that includes a new function to print a checkerboard pattern to the screen given input row and column size parameters.
9	Chapter Nine: Debugging and Exceptions • Logic Errors, Runtime Errors and Exceptions • Catching Exceptions • Finding Runtime Errors • The Eclipse Debugger	**Bug Hunt** – The student is presented with a program that contains a number of bugs. The student will use the Eclipse debugger and troubleshooting skills to identify and resolve each issue.

Week	Reading and Objectives	Labs
10	Chapter Ten: Introduction to OOP • Object-Oriented Concepts • Defining a Class • Public, Private, and Protected Classes	**Dog House** – The student will write their first multi-object program and observe the interaction between objects.
11	Chapter Eleven: Objects in Java • Constructors • Object Interfaces • Static Members	**Let's Go Racing!** – The student will create a RaceCar object and an IRacer object. Multiple RaceCar instances will be added to a provided RaceTrack object that knows how to run races through the IRacer interface.
12	Chapter Twelve: Graphical Java Programs • Java Swing • Creating a Simple Window • Event-Driven Programming • Layout Managers	**Phone Dialer** – The student's first Java Swing program will show a simple phone keypad and allow users to enter a phone number for display.
13	Chapter Thirteen: Swing Input Controls • Text and Numeric Input • List Input • Option Input	**Pizza Place** – The student will create a pizza ordering screen to demonstrate proper use of many common UI widgets (check boxes, radio buttons, list boxes, etc).
14	Chapter Fourteen: Arrays and Collections • Arrays (1D and 2D) • Java Lists and ArrayLists • Iterators	**Baseball Stats** – The student will use 1D arrays of integers and ArrayLists containing Player objects to insert, track and calculate baseball player batting statistics.

Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
15-16	Chapter Fifteen: Inheritance and Polymorphism • Learn about the "Jail Break!" game. • Base Classes and Derived Classes • Using References to Base and Derived Classes • Overriding Base Methods • The "Object" Base Class • Using Base Features from Derived Classes	**Game Pieces** – The student will create three derived classes (Deputy, Henchman, Kingpin) from an abstract base, in preparation for using these classes in the mid-term project. The classes are tested to ensure they meet the requirements using a provided test class.
17-18	Chapter Sixteen: Jail Break Project For the mid-term project the student will complete a game called "Jail Break" that is based on an old Viking board game. The student will create the abstract hierarchy of pieces (AbstractGamePiece, Deputy, Henchman, Kingpin) and write other logic to complete the game. The project consists of 6 guided lab steps that involve creating new classes, modifying existing code, and integrating with provided starter objects. Each guided step contains a checkpoint for testing to ensure code meets the requirements at each step.	**Building the Activity Starter** – Ensure the student can find and build the starter project. **Completing JailBreak.reset()** – Write logic to initialize the game board with pieces in the starting position. **Selecting Game Pieces** – Write game logic to allow selection and de-selection of game pieces. **Moving Game Pieces** – Write game logic (including virtual method overrides) to control game piece movement. **Capturing Game Pieces** – Write game logic to control game piece capturing. **Ending the Game** – Complete the end-of-game logic.

Week	Reading and Objectives	Labs
19	Chapter Seventeen: Math Functions in Java • Java Math Functions • The Binary Number System • Creating a MathFactory demonstration • Common Algorithms	**MathFactory Activity** – The student will expand the MathFactory lab to include decimal-to-binary conversion. **Algorithms Practice** – The student will gain experience writing their own simple algorithms.
20	Chapter Eighteen: File Access • Data Streams • Reading and Writing Text Data • Reading and Writing Binary Data	**Address CSV** – The student will write a program to convert a list of Address structures to a CSV file on disk, and then read that file back in again and re-populate the address list.
21	Chapter Nineteen: Sorting, Searching and Recursion • Recursion • Sorting Algorithms • Searching Algorithms	**Recursive Binary Search** – The student will write a binary search function to locate a number in a pre-sorted array.
22	Chapter Twenty: Program Efficiency • Common Algorithms • Algorithm Performance (Big-O) • Measuring Sorting Efficiency	**Comparison of Sorting Algorithms** – The student will implement timing and data-generation algorithms and measure the performance of 4 different sort routines with various numbers of elements.

Week	Reading and Objectives	Labs
23	Chapter Twenty-One: Vector and Bitmap Images • Screen Coordinates • Drawing Shapes • Drawing Images	**Sky Art** – The student will use recursion, vector graphics, and image graphics to generate a randomized cloudy sky scene.
24	Chapter Twenty-Two: Object Composition and Copying • Functional Decomposition • Composite Classes • Copying Objects	**Designing a Composite Class** – Students practice defining a composite class from smaller objects.
25	Chapter Twenty-Three: Computer Networking • Basic Networking • Network Topology • Network Addressing	**Animal Palace** – Students will use online tools to find images and store in a shared directory and class web page.
26	Chapter Twenty-Four: Software Engineering Principles • Design Processes and Teamwork • Java Doc • Testing Your Code	**Creating JavaDoc HTML** – The student will add JavaDoc comments to an earlier lab project and generate HTML output using the javadoc tool.

Week	Reading and Objectives	Labs
27-28	Chapter Twenty-Five: Team Project **Individual students may complete all phases of the team project by themselves, if desired.** The final project can be completed before or after the AP exam and the timeline scaled to fit available time. Student-driven labs will cover each phase of the software lifecycle. • Project Requirements • Project Design • Project Implementation • Project Testing	**Team Project Requirements** – Student teams will define their final project requirements. **Project Design** – Student teams will design their final projects. **Team Project Implementation** – Student teams will code their final project. **Team Project Testing** – Student teams will test their final project.
29-30, or as desired	Chapter Twenty-Six: Supplemental Labs This chapter describes the supplemental "exemplar" labs published by the College Board: • Magpie Lab • Picture Lab • Elevens Lab The 2014-2015 AP exam drops the GridWorld case study in favor of three new labs (Magpie, Picture, Elevens). None of the labs are required, but represent the type of work students should complete prior to taking the exam. Teachers may use any or all of the labs (in whole or part) in order to best meet their classroom needs.	**Magpie** – Guided activities in the Magpie Lab. **Picture Lab** – Guided activities in the Picture Lab. **Elevens** – Guided activities in the Elevens Lab. **Note: AP teachers must obtain the restricted lab material from the College Board by completing the course audit process. A pre-approved syllabus is available on our website.**

Week	Reading and Objectives	Labs
31, or as desired	Chapter Twenty-Seven: GridWorld Case Study This chapter contains a guided walkthrough of the older College Board "GridWorld" case study. It is no longer required on the AP exam, but remains an interesting example for students to consider. Teachers may use this lab (in whole or part) in order to best meet their classroom needs.	**GridWorld** – Guided activities in the GridWorld case study.
31-32	AP EXAM – PRACTICE TESTS, REVIEW, MAKE-UP WORK	Flexible time used to review and practice for the AP exam.
33	AP EXAM – EARLY MAY	
34+	After the exam, the class will work on completion / extension of Team Project or other fun teacher-driven activities. Teachers can assign any mixture of Supplemental Lessons and labs based on class interest. The following Supplemental Lessons are included: • Software Development Careers • Technical Writing • Stacks and Queues • Software Development Process • Exploring UML • Productivity Tools	Hands-on activities included in each supplemental lesson.