Java Programming (Abridged)

Course Syllabus and Planner

Updated July, 2019

Course Overview

The *Java Programming (Abridged)* curriculum is a one-year (two-semester) course covering topics typically found in **Computer Science I** or similar courses. This course has been aligned to specific course standards in a number of states.

This abridged course contains the same material as the full *Java Programming* course, minus some advanced chapters and AP-centric material. Other introductory programming courses are not required; students merely need to have typical computer usage skills prior to starting this course.

Teaching Strategies

The course material is designed to appeal to a variety of students, from traditional learners who thrive on written text to audio-visual students who enjoy a multi-media format. All content is delivered through an online system that allows students to work seamlessly both in the classroom and at home.

Labs and Programming Environment

Every chapter contains one or more hands-on programming labs where students will design or implement programs to demonstrate understanding of the lesson topics. Students will get the opportunity to work on individual and group projects and will experience all phases of a project lifecycle, including requirements, design, implementation, and testing.

The primary Java programming IDE is "Eclipse" (www.eclipse.org). The course contains detailed download, installation, and usage instructions. Alternative IDEs may be used at teacher discretion.

Course Planner

A typical school year consists of approximately 36 calendar weeks or 180 days of school. The course plan covers approximately 130 school days, with additional time allocated for review, make-up work, individual projects and Supplemental Lessons. Each "day" listed below represents one typical class period of 45 – 60 minutes, so students will typically work 3-5 hours per week. The suggested number of days factors in 1 day per lesson & quiz, 1 day per lab, and 1 day per test. Some classes may move faster or slower than the suggested pace.

Each chapter contains multiple lessons, quizzes and a chapter test in addition to the listed Lab assignments. Teachers may utilize Supplemental Lessons as desired to meet state standards or student interest. Team Projects and other assignments may be adjusted to fit the available time.

Days	Reading and Objectives	Labs
6	Chapter One: Understanding Computer Programming • A Survey of Computer Hardware • Introduction to Computer Software • Common Programming Languages • Computer Ethics • Computer Security	Class discussion and review of a sample EULA terms and conditions.
6	Chapter Two: Getting Started with Java • The Java Platform • Writing Your First Program • Building and Running from the Command Line • Java Classes and Packages	**Establish Development Environment** - Install JDK, create working directory, and practice submitting projects through the online interface. **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.

Days	Reading and Objectives	Labs
6	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.
5	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.
7	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.
5	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.

Days	Reading and Objectives	Labs
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.
5	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.
6	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.
5	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.

Days	Reading and Objectives	Labs
5	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.
6	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.
5	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).
5	Chapter Fourteen: Arrays and Collections • 1D Arrays • 2D Arrays • Java Lists • ArrayLists • Iterators and Enhanced for() Loops	**Gold Rush** – The student will write algorithms to populate and traverse a data structure that combines 2D arrays and ArrayLists.

Days	Reading and Objectives	Labs
8	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.
10	Chapter Sixteen: Jail Break Project For this 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.

Days	Reading and Objectives	Labs
6	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.
5	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.
5	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.
5	Chapter Twenty: 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.

Days	Reading and Objectives	Labs
10+	Chapter Twenty-One: Team Project

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. |
| **Up to 30** | Supplemental Lessons, make-up work, review

Teachers can assign any mixture of Supplemental Lessons and labs as required by state standards or based on class interest. Three full chapters of supplemental material are available, each with either a lab or a quiz in addition to the lesson text.

Suppl. Chapter 1: Enrichment Topics (7 lessons)

Suppl. Chapter 2: Software and Industry (4 lessons)

Suppl. Chapter 3: Computers and Modern Society (4 lessons) | |