Teaching Tomorrow's Technology.
CompuScholar

Course Syllabus and Planner for Tennessee

Coding I
&
Coding II

Updated March, 2019

Course Overview

The CompuScholar *Java Programming (AP)* curriculum is normally a one-year (two-semester) course covering all required topics in the **AP Computer Science A** exam. By popular request, this document explores a scheduling pattern and identifies supplements needed to comply with both the Tennessee "**Coding I**" and the "**Coding II**" standards as a two-year sequence.

Course Planner

To complete both Tennessee course standards over a two-year sequence, the normal 1-year pacing for CompuScholar's *Java Programming (AP)* course should be extended with a number of supplemental lessons and activities to meet the additional standards.

- **Year 1** – **Coding I (CI) -** can be completed with the first 2/3 of CompuScholar's *Java Programming (AP)* plus supplemental material

- **Year 2** - **Coding II (CII) -** can be completed with the second 1/3 of CompuScholar's *Java Programming (AP)* plus supplemental material

The following tables describe how to meet all **Coding I** and **Coding II** standards over a 2-year sequence using CompuScholar's *Java Programming (AP)* as the primary resource. Where specific **CI** or **CII** standards are not met, the source is identified as "Teacher Supplements", which should be provided by the teacher.

Coding I – Year 1

Week	Reading and Objectives	Source / Comments
1	Chapter One: Understanding Computer Programming	Java Programming (AP)
2	Chapter Two: Getting Started with Java	Java Programming (AP)
3	Chapter Three: The Eclipse IDE	Java Programming (AP)
4	Chapter Four: Data Types and Variables	Java Programming (AP)
5	Chapter Five: Working With Strings	Java Programming (AP)
6	Chapter Six: User Input	Java Programming (AP)
7	Chapter Seven: Basic Flow Control	Java Programming (AP)
8	Chapter Eight: Writing Methods	Java Programming (AP)
9	Chapter Nine: Debugging and Exceptions	Java Programming (AP)
10	Chapter Ten: Introduction to OOP	Java Programming (AP)
11	Chapter Eleven: Objects in Java	Java Programming (AP)
12	Chapter Twelve: Graphical Java Programs	Java Programming (AP)
13	Chapter Thirteen: Swing Input Controls	Java Programming (AP)
14	Chapter Fourteen: Arrays and Collections	Java Programming (AP)
15	Supplemental Lesson 4 (Software Development Process) Supplemental Lesson 1 (Software Development Careers)	Java Programming (AP)
16-17	Mid-Term Project	Java Programming (AP)
18-19	Chapter Fifteen: Inheritance and Polymorphism	Java Programming (AP)
20-21	Chapter Sixteen: Jail Break Project	Java Programming (AP)
22	Chapter Seventeen: Math Functions in Java	Java Programming (AP)
23	Chapter Eighteen: File Access	Java Programming (AP)

24	Supplemental Chapter: Security and Privacy - Use Digital Savvy material as a basis	CI #15 – Teacher supplements and CompuScholar: Digital Savvy Chapter 8
25-30	Final Project	Similar to mid-term, expanded scope

Coding II – Year 2

Note: The **Coding II** course standards largely revolve around a collection of group projects and development of "soft" skills (teamwork, SDLC process). Many district teachers prefer individual approaches to address most line-item requirements. Therefore, the timeline below leaves substantial space for teachers to fill using their preferred approach. We encourage using the last 1/3 of the *Java Programming (AP)* material to continue to build Java skills and making the entire course available to students as a reference and catch-up source.

Week	Reading and Objectives	Source / Comments
1-4	Review of Coding I Java skills	Java Programming (AP) Chapters 1 – 16 as needed
5	Course-Long Programming Project Kickoff	CII #20 - Teacher supplements
6	Chapter Nineteen: Sorting, Searching and Recursion	Java Programming (AP)
7	Chapter Twenty: Program Efficiency	Java Programming (AP)
8	Chapter Twenty-One: Vector and Bitmap Images	Java Programming (AP)
9	Chapter Twenty-Four: Software Engineering Principles	Java Programming (AP)
10	Supplemental Chapter: Evaluate Other Languages and IDEs - Briefly done with hour of code / python or other easily accessible language	CII #1 – Teacher supplements

11	Expanded SDLC Research - Expand Supplemental Lesson 4 with teacher supplements	CII #4 - Teacher supplements and Java Programming (AP) Supplemental Chapter 2, Lesson 1
12-32	Teacher discretion and approach while conducting group projects: CII #5 – choose/defend lifecycle and technology CII #6 – Research best practices for SDLC model CII #7 – use requirements management tool CII #8 – choose/defend language CII #9 – identify data processing approach CII #10 – data management plan CII #13 – user documentation CII #15 – choose/defend programming approach CII #16 - design an app for a mobile device – suggest using AppInventor or similar online tool CII #18 – analyze and flowchart someone else's code	Teacher supplements