



Java Programming For AP Computer Science A

Course Syllabus and Planner

Draft – Subject to Change!

Course Overview

CompuScholar's **Java Programming** curriculum is a one-year (two-semester) course that covers all topics found in the College Board **AP Computer Science A** course description. 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:

<https://www.compuscholar.com/schools/courses/java/>

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

Java Programming can also be used as a non-AP, introductory computer science course. For details, please see our Syllabus and Planner guide for non-AP CS A teachers.

Course Material

The course consists of the following student-facing elements:

- **Instructional Videos** – optional (not required), but enjoyed by many students as an audio-visual introduction and re-enforcement of the lesson topics.
- **Lesson Text** – required reading, contains full topic details and live coding exercises
- **Quizzes and Exams** – multiple choice and automatically graded by our system
- **Chapter Activities** – hands-on projects, submitted for a grade

Teachers additionally have access to:

- **Teacher's Guides** – for each lesson, with suggested classroom discussion questions
- **Quiz and Exam Answer Keys** – PDFs for quick reference
- **Activity Solution Guides** – fully coded activity solutions for each chapter activity



Programming Environment and Device Requirements

CompuScholar provides an in-browser Java coding environment. This online feature may be used by students to complete all exercises and activities in all required AP chapters. When using our online coding environment:

- **No local software installation is needed to prepare for the AP exam.**
- **The AP material can be completed from any web browser on any device (including Chromebooks and tablets).**

Later, optional chapters contain a mixture of activities. AP teachers may select any of these topics for students after the AP exam. Some optional activities can be done in CompuScholar's online environment, while others are completed using an external IDE.

When needed, we recommend a locally installed JDK and Eclipse platform for an external IDE (see chapters 28 and 29 for instructions). Teachers may also select any other locally installed or online IDE. **Device requirements for your optional, external IDE depend on the IDE selected.**

Project Grading

Each chapter normally contains one or more hands-on, graded activities. The activities in **all required AP chapters are fully auto-graded by our system.** Teachers have complete control over the auto-graded results.

Some activities in later, optional chapters are free-form (creative) or completed in an external IDE. The teacher is responsible for grading those creative or external projects.

Course Navigation

Chapter 1 contains computing, ethics and security topics recommended (but not tested) by AP CS A and required by many state and national computer science curriculum standards. AP teachers may opt to complete Chapter 1 in sequence, skip it entirely, or return to the topics after the AP CS A exam.



Chapters 2 – 24 should be completed in sequence and cover all required topics on the AP CS A exam, plus certain other highly recommended software skills. These chapters include substantial, hands-on lab work in excess of the 20-hour minimum AP requirement. The mid-term project in Chapter 17 may be omitted or postponed, if desired (no new skills).

Typical classes will finish all required AP content prior to the exam administration in May. We recommend using remaining time before the exam to review the College Board's published practice exams and any other external source of practice FRQ and multiple choice problems.

Chapters 25 – 33 contain optional topics that are not required for AP CS A. Teachers may review and select any of these optional topics for students as time permits after the AP CS A exam. Some optional chapters require use of an external IDE and/or will be graded by the teacher.

Supplemental Chapters 1 – 4 contain a variety of enrichment topics that may be required by individual states to satisfy requirements for other coding or digital literacy courses. AP teachers may optionally select any of these topics for students, time permitting.

Course Planner

The following pages contain a suggested timeline for completing course content. Correlations to the AP CS A recommended Unit Sequence are highlighted. Additional, detailed mappings to AP CS A Learning Objectives and Essential Knowledge (LOEKs) can be found on our [course description page](#).

A typical school year consists of 36 calendar weeks or 180 days of school. After completing the first 24 chapters, most classes will have several weeks left for AP exam prep, make-up work and optional topics. Teachers can select from optional topics before and after the exam, as time permits.

Each "day" listed below represents one typical day or class period of 45 – 60 minutes. In most cases, we anticipate students will complete one lesson per day (including the quiz), 1 day per lab, and 1 day per chapter test. Some classes may move faster or slower than the suggested pace.



Semester 1 Timeline

Days	CompuScholar Chapter	AP CS A Unit Sequence
6	Chapter 1: Computing Concepts <ul style="list-style-type: none">* Evolution of Computers* Computer Hardware* Computer Software* Computer Ethics* Computer Security	General curricular requirements (N/A on AP Exam) Schedule as time permits

6	Chapter 2: Getting Started with Java <ul style="list-style-type: none">* Common Programming Languages* The Java Platform* Writing Your First Program* Help and Reference Documentation ACTIVITY: Shopping List	UNIT 1: Primitive Types 1.1 Why Programming? Why Java? 1.2 Variables and Data Types 1.3 Expressions and Assignment Statements 1.4 Compound Assignment Operators (shortcuts) 1.5 Casting and Ranges of Variables
5	Chapter 3: Data Types and Variables <ul style="list-style-type: none">* Primitive Data Types* Variables* Printing Data ACTIVITY: Treasure Map	
5	Chapter 4: Working with Numbers <ul style="list-style-type: none">* Simple Math Operations* Compound Assignments and Shortcuts* Type Casting and Truncation ACTIVITY: Magic Math	



7	Chapter 5: Introducing Objects <ul style="list-style-type: none">* Java Classes* Reference Variables and Strings* Properties and Constructors* Calling Methods* User Input with Scanner ACTIVITY: Sketch Robot	UNIT 2: Using Objects <ul style="list-style-type: none">2.1 Objects: Instances of Classes2.2 Creating and Storing Objects (Instantiation)2.3 Calling a Void Method2.4 Calling a Void Method with Parameters2.5 Calling a Non-void Method2.6 String Objects: Concatenation, Literals, and More2.7 String Methods2.8 Wrapper Classes: Integer and Double2.9 Using the Math Class
5	Chapter 6: Working with Strings <ul style="list-style-type: none">* Comparing Strings* Common String Operations* Formatting and Building Strings ACTIVITY: String Theory	
5	Chapter 7: Numbering Systems and Java Math <ul style="list-style-type: none">* Java Wrapper Classes & Numeric Conversion* Numbers in Binary, Octal and Hex* Java Math Class ACTIVITY: Math Factory	
6	Chapter 8: Logic and Decision-Making <ul style="list-style-type: none">* Logical Expressions and Relational Operators* Making Decisions with if()* Using "else-if" and "else"* The "switch" Statement ACTIVITY: Banking System	UNIT 3: Boolean Expressions and if Statements <ul style="list-style-type: none">3.1 Boolean Expressions3.2 if Statements and Control Flow3.3 if-else Statements3.4 else if Statements3.5 Compound Boolean Expressions3.6 Equivalent Boolean Expressions3.7 Comparing Objects
5	Chapter 9: More Complex Logic <ul style="list-style-type: none">* Comparing Objects and References* Compound Expressions* Boolean Algebra and Truth Tables ACTIVITY: Wild Card	



Teaching Tomorrow's Technology.

CompuScholar

5	Chapter 10: Handling Exceptions <ul style="list-style-type: none">* Understanding Exceptions* Catching Exceptions* Validating User Input ACTIVITY: Calculator Madness	Highly recommended skills as students begin to produce more complex code.
4	Chapter 11: Debugging <ul style="list-style-type: none">* Finding Runtime Errors* Debugger Concepts ACTIVITY: Bug Hunt	

6	Chapter 12: Iteration <ul style="list-style-type: none">* For Loops* While Loops* Continue, Break and Return* Nested Loops ACTIVITY: Fun Factorials	UNIT 4: Iteration 4.1 while Loops 4.2 for Loops 4.3 Developing Algorithms Using Strings
6	Chapter 13: Algorithms <ul style="list-style-type: none">* Designing with Flowcharts* Writing Pseudocode* Common Mathematical Algorithms* Common String Algorithms ACTIVITY: Meal Time	4.4 Nested Iteration 4.5 Informal Code Analysis



7	Chapter 14: Creating Java Classes <ul style="list-style-type: none">* Object-Oriented Concepts* Defining Classes and Packages* Class Properties* Constructors* Class Methods ACTIVITY: Dog House	UNIT 5: Writing Classes 5.1 Anatomy of a Class 5.2 Constructors 5.3 Documentation with Comments 5.4 Accessor Methods 5.5 Mutator Methods 5.6 Writing Methods 5.7 Static Variables and Methods 5.8 Scope and Access 5.9 this Keyword 5.10 Ethical and Social Implications of Computing Systems (Not on AP Exam) - See Chapter 1
7	Chapter 15: Working with Methods <ul style="list-style-type: none">* Documentation and Design* Variable Scope and Access* Data Encapsulation* Method Overloading* Object Interfaces ACTIVITY: Let's Go Racing!	
5	Chapter 16: Static Concepts <ul style="list-style-type: none">* Static Properties* Static Methods* Static, Object and "this" References ACTIVITY: Art School	
4	Chapter 17: Mid-Term Project <ul style="list-style-type: none">* Introducing the "Remote Control" Project ACTIVITY: Creating the Schedule ACTIVITY: Building a Television ACTIVITY: Defining the Remote	Schedule as time permits (no new skills)
84 - 94	Total Days (depending on scheduling of Chapters 1 and 17)	



Semester 2 Timeline

Days	CompuScholar Chapter	AP CS A Unit Sequence
7	Chapter 18: 1D Arrays <ul style="list-style-type: none">* Array Concepts* Array Traversal* Iterators and Enhanced for() loops* Array Algorithms* More Array Algorithms ACTIVITY: Whack-A-Mole	UNIT 6: Array <ul style="list-style-type: none">6.1 Array Creation and Access6.2 Traversing Arrays6.3 Enhanced for Loop for Arrays6.4 Developing Algorithms Using Arrays
6	Chapter 19: Lists and ArrayLists <ul style="list-style-type: none">* Java Lists* ArrayLists* Iterators and Enhanced for() Loops* Algorithms with ArrayLists ACTIVITY: Train Yard Jumble	UNIT 7: ArrayList <ul style="list-style-type: none">7.1 Introduction to ArrayList7.2 ArrayList Methods7.3 Traversing ArrayLists7.4 Developing Algorithms Using ArrayLists
7	Chapter 20: Searching and Sorting <ul style="list-style-type: none">* Bubble Sort* Selection Sort* Insertion Sort* Sequential and Binary Searches ACTIVITY: Ducks in a Row	<ul style="list-style-type: none">7.5 Searching7.6 Sorting7.7 Ethical Issues Around Data Collection (see Chapter 1)
6	Chapter 21: 2D Arrays <ul style="list-style-type: none">* 2D Arrays* Traversal and Ordering* Array of Arrays* 2D Array Algorithms ACTIVITY: Gold Rush	UNIT 8: 2D Array <ul style="list-style-type: none">8.1 2D Arrays8.2 Traversing 2D Arrays



5	Chapter 22: Inheritance * Superclass and Subclass Concepts * Subclass Constructors * Using Superclass and Subclass References ACTIVITY: Lab Rats	UNIT 9: Inheritance 9.1 Creating Superclasses and Subclasses 9.2 Writing Constructors for Subclasses 9.3 Overriding Methods 9.4 super Keyword 9.5 Creating References Using Inheritance Hierarchies 9.6 Polymorphism 9.7 Object Superclass
6	Chapter 23: Polymorphism * Overriding Superclass Methods * Abstract Classes and Methods * Using Superclass Features with "super" * The "Object" Superclass ACTIVITY: Social Ladder	
5	Chapter 24: Recursion * Recursion * Recursive Binary Search * Merge Sort ACTIVITY: File Explorer	UNIT 10: Recursion 10.1 Recursion 10.2 Recursive Searching and Sorting
42	Total Days in Semester 2 (all required AP CS A topics complete at this point)	

Classes who complete the first 24 chapters at this point have spent approximately 134 days and covered all required AP CS A topics. Remaining class time should be spent in preparation for the AP exam and in optional, teacher-selected topics from Chapters 25 - 33, Supplemental Chapters, and any earlier chapters deferred until after the exam.

Please see below for information on the **optional chapters and Supplemental topics**.



The following table suggests the timeline needed for each **optional or supplemental chapter**, along with notes as to the programming environment and grading approach. There are more “optional” chapters available than students can complete in a single year, so teachers can pick topics as time permits!

Days	CompuScholar Chapter	Notes
5	Chapter 25: File Access <ul style="list-style-type: none">* Data Streams* Reading and Writing Text Data* Reading and Writing Binary Data ACTIVITY: Address CSV	CompuScholar online environment, project auto-graded by our system
5	Chapter 26: Object Composition and Copying <ul style="list-style-type: none">* Functional Decomposition* Composite Classes* Copying Objects ACTIVITY: Designing a Composite Class	Teacher-graded project
10-15	Chapter 27: Team Project <ul style="list-style-type: none">* Design Processes and Teamwork* Requirements and Design Documents ACTIVITY: Team Project Requirements ACTIVITY: Project Design ACTIVITY: Team Project Implementation <ul style="list-style-type: none">* Testing Your Code ACTIVITY: Team Project Testing	CompuScholar online environment or external IDE, teacher-graded project
3	Chapter 28: Running Java Locally <ul style="list-style-type: none">* Installing the JDK* Local Source Code* Building and Running from the Command Line	“How-to” chapter to create local development environment
4	Chapter 29: The Eclipse IDE <ul style="list-style-type: none">* Introducing Eclipse* Eclipse Java IDE Walk-through* Creating an Eclipse Project* The Eclipse Debugger	“How-to” chapter to install and use a local IDE
6	Chapter 30: Graphical Java Programs <ul style="list-style-type: none">* Java Swing* Creating a Simple Window* Event-Driven Programming	Requires external IDE (e.g. Eclipse) with Java Swing support. Teacher-graded projects.



	* Layout Managers ACTIVITY: Phone Dialer	
5	Chapter 31: Swing Input Controls * Text and Numeric Input * List Input * Option Input ACTIVITY: Pizza Place	
5	Chapter 32: Vector and Bitmap Images * Screen Coordinates * Drawing Shapes * Drawing Images ACTIVITY: Sky Art	
4	Chapter 33: Program Efficiency * Algorithm Performance (Big-O) * Measuring Sorting Efficiency ACTIVITY: Comparison of Sorting Algorithms	External IDE, teacher-graded project
12	Supplemental Chapter 1: Enrichment Topics	See individual lessons and activities for programming environment and grading approach.
8	Supplemental Chapter 2: Software and Industry	Offline work, teacher-graded projects
4	Supplemental Chapter 3: Computers and Modern Society	Offline work, teacher-graded projects
6	Supplemental Chapter 4: Computer Networking	Offline work, teacher-graded projects