

Arkansas Software Development Syllabus

High School - One Year (150 hours)

Course Overview and Goals

Arkansas Software Development is a year-long, Java-based course that takes students from the fundamentals of programming through the practices of professional software development. Students begin by writing Java programs with variables, methods, conditionals, loops, and their own classes, then advance to data structures, object-oriented design, and algorithms. The course also explores the wider profession, including the software development life cycle, testing, version control, databases and SQL, security and accessibility, and emerging technologies such as artificial intelligence and cloud computing. By the end of the course, students can design, build, test, and present a complete software project and assemble a portfolio that showcases their work.

Course Background and Resources

Prerequisites

This course is designed for high school students and assumes no prior programming experience. A foundation in basic algebra is helpful. It can serve as a student's first formal computer science course or follow an introductory course in the pathway.

Learning Environment

The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each module of the course is broken down into lessons. Lessons may be composed of short video tutorials, interactive learning pages, explorations, simulations, free-response prompts, coding exercises, and quizzes. Each module ends with a comprehensive quiz that assesses students' mastery of that module's material.

Programming Environment

Students write and run Java programs in the browser using the CodeHS editor.

Technology Requirements

You can find the basic technology requirements for using the CodeHS platform in the [Technical Set-Up Guide](#).

To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: [Arkansas Software Development Course Links](#)

More Information

Browse the content of this course at <https://codehs.com/course/28572/explore>

Course Breakdown

Module 1: The Software Profession (1 week / 5 hours)

Students explore what it means to work in software, including developer roles, teamwork, professional communication, and career pathways such as internships and leadership.

Topics Covered	● Developer roles ● Software teams ● Collaboration ● Success traits ● Technical communication ● Internships ● CS careers ● Leadership
Example Assignments	● Which Role Fits You? ○ Students analyze software developer roles and reflect on which best fits their strengths and interests. ● Internships Exploration ○ Students research real computer science internships and summarize one they find interesting. ● Contact an Organization ○ Students draft a professional outreach email to a company or professional. ● Technical Writing Practice ○ Students write a clear technical summary about the history of Java.

Module 2: Using Objects and Methods (4 weeks / 20 hours)

Students learn the fundamentals of Java, including variables, data types, expressions, input and output, methods, and how to use objects and the String class.

Topics Covered	• Variables & data types • Expressions & output • Input • Casting • Methods • Math class • Objects • String manipulation
Example Assignments	• Tip Calculator ◦ Students compute a restaurant tip and total from user input. • Ice Cream Shop ◦ Students read user input and produce a formatted order confirmation. • MLA Citation Generator ◦ Students use String methods to assemble a correctly formatted citation. • Movie Ratings ◦ Students apply casting and the range of variables to work with numeric ratings.

Module 3: Selection and Iteration (3 weeks / 15 hours)

Students write decision-making and repetition logic using boolean expressions, conditionals, and loops, and learn a systematic process for debugging their programs.

Topics Covered	• Boolean expressions • if statements • Nested conditionals • Compound booleans • Loops • Nested iteration • Debugging
-----------------------	--

	● Run-time analysis
Example Assignments	● Sweet or Unsweet? ○ Students build a drink-order program that responds to user choices with conditionals. ● Temperature Classifier ○ Students use nested if statements to classify temperatures into categories. ● Find the Median ○ Students use conditional logic to determine the median of three integers. ● Analyzing a Guessing Game Algorithm ○ Students trace and analyze an algorithm that uses selection and iteration.

Module 4: Class Creation (3 weeks / 15 hours)

Students design and build their own Java classes, including instance variables, constructors, methods, scope, and encapsulation using accessor and mutator methods.

Topics Covered	● Program design ● Anatomy of a class ● Constructors ● Writing methods ● Scope & access ● this keyword ● Accessors ● Mutators
Example Assignments	● Dog Class ○ Students write a class with instance variables and a constructor. ● Song Class ○ Students use the this keyword and mutator methods to update object state. ● Public Library System ○ Students design a multi-class system with a diagram before writing code. ● Order Up ○ Students use parameters and return values to pass data between methods with controlled access.

Module 5: Data Collections (5 weeks / 25 hours)

Students store and process collections of data using arrays, ArrayLists, and text files, and implement searching, sorting, and recursive algorithms.

Topics Covered	• Arrays • Array traversals • Text files • Exceptions • Wrapper classes • ArrayLists • Searching & sorting • Recursion
Example Assignments	• Warehouse Inventory ◦ Students traverse an array to track and update inventory data. • Clean the Roster File ◦ Students read a text file and handle messy, real-world data. • Make the Parser Robust ◦ Students add exception handling so a file parser survives bad input. • FitBit Data Set: Sleep Analysis Algorithm ◦ Students design an algorithm that analyzes a real data set.

Module 6: OOP Extensions & Architecture (4 weeks / 20 hours)

Students extend classes with inheritance and polymorphism and study how design patterns and software architecture organize larger programs.

Topics Covered	• Inheritance • Subclass constructors • Overriding methods • super keyword • Polymorphism • Object superclass • Design patterns • Architecture
-----------------------	--

Example Assignments	● Person / Student Object ○ Students create and use objects from a parent class and its subclass. ● Electric Cars ○ Students create an ElectricCar subclass of an existing Car class. ● Bank Accounts ○ Students create an ElectricCar subclass of an existing Car class. ● Online Companies ○ Students model a class hierarchy that reflects real-world relationships.
----------------------------	--

Module 7: SDLC, Testing, Version Control & Professional Practice (2 weeks / 10 hours)

Students explore what it means to work in software, including developer roles, teamwork, professional communication, and career pathways such as internships and leadership.

Topics Covered	● Software life cycle ● Requirements & planning ● Software testing ● Version control ● Code review ● Git & GitHub ● Career prep ● Portfolios & resumes
Example Assignments	● Write the User Stories ○ Students plan a feature's requirements by writing user stories. ● TDD: Tests First ○ Students practice test-driven development by writing tests before code. ● Conduct a Code Review ○ Students review a program and give structured, constructive feedback. ● Git Workflow Walkthrough ○ Students work through a real branching and merging Git workflow.

Module 8: Data Systems & APIs (2 weeks / 10 hours)

Students work with the command line, relational databases, and SQL, and build programs that read and use data.

Topics Covered	• Command line • Databases • SQL basics • Structuring data • Querying • Filtering queries • Data-driven programs • APIs
Example Assignments	• Your first <code>SELECT</code> statement ◦ Students write a basic SQL query to retrieve data from a table. • The Weasleys ◦ Students filter query results using WHERE conditions. • Schema Brainstorm ◦ Students design a database schema for a given scenario. • File Organizer ◦ Students use command-line commands to back up and organize files.

Module 9: Security, Privacy, Licensing & Accessibility (2 weeks / 10 hours)

Students examine operating systems, software licensing, application security, networking, copyright, data ethics, and web accessibility.

Topics Covered	• Operating systems • Software licenses • Application security • System administration • Networking • Copyright • Data ethics • Accessibility
Example	• Which would you choose?

Assignments	○ Students compare operating systems and justify a choice for a use case. ● Security Baseline ○ Students review a security baseline document and apply its objectives. ● Cite! ○ Students apply copyright and attribution rules by citing sources correctly. ● Find the API's Flaws ○ Students analyze an API to identify security weaknesses.
--------------------	---

Module 10: Emerging Technologies (1 week / 5 hours)

Students investigate current and emerging technologies, including cloud computing, artificial intelligence and machine learning, and blockchain.

Topics Covered	● Cloud computing ● Modern dev approaches ● Artificial intelligence ● Machine learning ● Neural networks ● Blockchain
Example Assignments	● Infographic: Cloud Technologies ○ Students create an infographic that explains cloud computing services. ● Monolith or Microservices? ○ Students evaluate and compare software architecture approaches. ● Machine Learning and NN Reflection ○ Students analyze how machine learning and neural networks work. ● Evaluate a Low-Code Tool ○ Students assess a modern low-code development tool and its trade-offs.

Module 11: Final Project (3 weeks / 15 hours)

Students plan, build, test, present, and publish a capstone software project and assemble a professional portfolio.

Topics Covered	● Project planning ● Capstone build ● User testing ● Evaluation ● Presentation ● Portfolio
Example Assignments	● Build Your Capstone ○ Students design and build an original capstone program of their own. ● User Testing 1 ○ Students gather user feedback on their project and respond to it. ● Final Reflection & Presentation Write-Up ○ Students present their finished project and reflect on the process. ● Portfolio Plan + Publish ○ Students assemble and publish a professional portfolio of their work.