



CodeHS

Introduction to Java Semester 1A Course Syllabus

The CodeHS Intro to Java Semester A and Semester B courses are aligned to all College Board seven curriculum requirements extensively as shown in the table below. However, some more advanced topics, like recursion and the various sorting algorithms have been omitted. Therefore, these courses do NOT prepare students for the AP CS A exam. The curriculum requirements laid out by the College Board are:

- ❖ CR1: Teaches students to design and implement computer-based solutions to problems.
- ❖ CR2a: Teaches students to use and implement commonly used algorithms.
- ❖ CR2b: Teaches students to use commonly used data structures.
- ❖ CR3: Teaches students to select appropriate algorithms and data structures to solve problems.
- ❖ CR4: Teaches students to code fluently in an object-oriented paradigm using the programming language Java.
- ❖ CR5: Teaches students to use elements of the standard Java library.
- ❖ CR6: Includes a structured-lab component composed of a minimum of 20 hours of hands-on lab experiences.
- ❖ CR7: Teaches students to recognize the ethical and social implications of computer use.

Course Overview and Goals

The CodeHS Intro to Java Semester A course is a semester-long course designed to help students master the basics of Java. It is the first course in a two course sequence and should be completed before the Intro to Java Semester B course. All learning materials and resources teachers and students need for a successful year-long Java course can be found on the CodeHS website.

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 unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 50 hours of hands-on programming practice in total. [CR6] Several units have free response questions that have students consider the applications of programming and incorporate examples from their own lives.

Programming Environment: Students write and run Java programs in the browser using the CodeHS editor. [CR1] [CR6]

Quizzes: At the end of each unit, students take a summative multiple choice unit quiz that assesses their knowledge of the Java concepts covered in the unit. Included in each lesson is a formative short multiple choice quiz.

Prerequisites

There are no official prerequisites for the CodeHS Intro Java Semester A course, however we recommend that students take our Introduction to Computer Science prior to Intro to Java (more info at codehs.com/library). Students who have completed our Intro to CS course will be able to apply knowledge of concepts covered in the Intro course to the more advanced setting of the Intro to Java course. It is also expected that students know basic English and algebra. Students should be comfortable with functions and function notation, such as $f(x) = x + 2$ and $f(x) = g(h(x))$.

Course Breakdown

Unit 1: Introduction to Programming in Java with Karel the Dog (3 weeks)

Objectives / Topics Covered [CR1]	<ul style="list-style-type: none">• Commands• Defining vs. Calling Methods• Designing methods• Program entry points• Control flow• Looping• Conditionals• Classes• Commenting code• Preconditions and Postconditions• Top Down Design
Assignments / Labs [CR1] [CR6]	<ul style="list-style-type: none">• 34 Karel Programming Exercises in total• Program-specific tasks for Karel the Dog<ul style="list-style-type: none">◦ Example Exercise: Maze Karel

Karel is stuck in a maze. Help him escape and find the tennis ball at the end.

Your job is to give commands to Karel to help navigate the maze and end up on the tennis ball.

Karel should end up facing East.

- Teach Karel new commands like `turnRight()` or `makePancakes()`

- Example Exercise:

Pancakes

Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes.

Create a method called `makePancakes()` to help Karel solve this problem.

- Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design

- Example Exercise:

The Two Towers

In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high.

At the end, Karel should end up on top of the second tower, facing East.

- Using control structures and conditionals to solve general problems

- Example Exercise:

Random Hurdles

Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

- Example Exercise:

Super Cleanup Karel

Karel's world is a complete mess. There are tennis balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing east, and should clean up all of the tennis balls in the world. This program should be general enough to work on any size world with tennis balls in any locations.