



CodeHS

Introduction to Java Semester 2A 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 B course is a semester-long course designed to help students master the basics of Java. It is the second course in a two course sequence and should be completed after the Intro to Java Semester A course. All learning materials and resources teachers and students need for a successful semester-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

Students should complete the Intro to Java Semester A course prior to starting this 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: Classes and Object Oriented Programming (8 weeks)

Objectives / Topics Covered [CR1] [CR4] [CR5]	<ul style="list-style-type: none">• Using classes as a client• Classes vs Objects• Class methods• Instance variables• Constructors• Visibility• Information hiding• <code>this</code>• <code>static</code>• <code>super</code>• The Java Math class and methods (<code>abs</code>, <code>pow</code>, <code>sqrt</code>, <code>sin</code>, <code>cos</code>)• Creating random values with the CodeHS Randomizer class• Designing classes• Creating classes• Getter and setter methods• Inheritance• Method overloading
--	---

	<ul style="list-style-type: none"> • Local variables and scope • Comparing objects vs primitive types • Abstract classes • packages • Polymorphism • Interfaces • Modifying classes to implement interfaces • Object is the superclass of all classes
Assignments / Labs [CR1] [CR4] [CR5] [CR6]	<ul style="list-style-type: none"> • Several programming exercises to master each of the topics above. 35 exercises in total. • Examples <ul style="list-style-type: none"> ○ Using the Student Class In this program we have a <code>Student</code> class in <code>Student.java</code> and a tester program at <code>StudentTester.java</code>. If you open up <code>StudentTester.java</code> you will see we have a bit of code there already. We've created two new students, Alan and Ada. We create a <code>Student</code> instance by calling the constructor and passing in the first name, last name, and grade level as an integer. Your task is to create a <code>Student</code> with your information! Once you have created the <code>Student</code>, print it out to the console. ○ Design and implement a <code>Fraction</code> class from scratch, including a constructor, getter and setter methods, a <code>toString</code> method, and methods to add, subtract, and multiply by other <code>Fraction</code> objects. ○ Implement a <code>RockPaperScissors</code> class with a <code>getWinner(String user, String computer)</code> method that allows a user to play the game Rock, Paper, Scissors against a computer that picks moves randomly. ○ Add an abstract method to an existing <code>Shape</code> class called <code>public abstract double getPerimeter()</code> and implement this method on each of the <code>Shape</code> subclasses, <code>Square</code>, <code>Rectangle</code>, <code>Pentagon</code>, and <code>Circle</code> ○ Fun with Solids Given the <code>Solid</code> abstract class, extend it with: <code>Pyramid</code> <code>Cylinder</code> <code>RectangularPrism</code> <code>Sphere</code> Make sure to create the constructor, <code>volume</code> and <code>surfaceArea</code> methods for each class (the <code>Math</code> class

	<p>will come in handy).</p> <p>Also extend <code>RectangularPrism</code> with <code>Cube</code>.</p> <ul style="list-style-type: none"> ○ Modify the <code>Fraction</code> class to implement the <code>Comparable</code> interface
--	--

Optional Supplemental Materials

<p>Objectives / Topics Covered</p> <p>[CR1] [CR4] [CR7]</p>	<ul style="list-style-type: none"> ● Extra practice with Java concepts <ul style="list-style-type: none"> ○ String processing ○ Recursion ○ Designing Classes ○ Arrays and <code>ArrayLists</code> ○ Searching and sorting algorithms ● File reading / writing ● The Java <code>Scanner</code> class ● The Java <code>BufferedReader</code> and <code>BufferedWriter</code> classes ● Running Java programs outside of the browser ● Running Java programs from the command line ● The Java <code>main</code> method ● Computing in Context <ul style="list-style-type: none"> ○ Understand computer ethics such as acceptable use policies, copyright, intellectual property, privacy, and the implications of developing software used by real people in real life situations
<p>Assignments / Labs</p> <p>[CR1] [CR4] [CR6] [CR7]</p>	<ul style="list-style-type: none"> ● Several additional exercises and advanced projects covering the topics listed above