CodeHS

AP Computer Science A (AP Java)
Common Core State Standards for Math
Standards Alignment Overview

The AP CS A (Java) course is an introductory college-level computer science course introducing the basics of programming with Karel the Dog and Java, object-oriented programming, data structures, and algorithms and recursion. Important Note: The AP CS A (Java) course is aligned to the College Board Framework, but also supports the Common Core State Standards for Mathematics.

Standards for Mathematical Practice

1. **Make sense of problems and persevere in solving them.**
   Errors are opportunities to learn from mistakes. Students continually test programs that that they build using Karel the Dog and Java, and these programs will invariably have bugs. Students are encouraged during testing to start by explaining to themselves the context the problem they are trying to solve and looking for entry points to its solution. They plan a solution pathway rather than jumping into a solution attempt with the issues they are experiencing. They persevere in solving them by using various debugging strategies. Debugging programs is a substantial focus in the AP Java course.

   Troubleshooting computing systems is a core concept of the AP CS A (Java) course as well. Computing systems might not work as expected because of problems in the software. Students are expected to identify problems in their programs and fix them.

2. **Reason abstractly and quantitatively.**
   Nearly every computer science course deals with various levels of abstraction, and the AP CS A (Java) course is no exception. In AP Java, students work with abstractions continually because of the object-oriented programming (OOP) nature of Java. Within designing their own classes, using abstract classes, interfaces, polymorphism and data structures, students experience the ultimate in abstracting within a programming language.

   Algorithms and programming are central to the AP CS A (Java) course. Students learn the core principles of developing their own algorithms and implementing them in the Java programming language. Algorithms, variables, control, modularity, classes, object-oriented programming, recursion, and program development are all taught in this

course. Students create their own Java methods and classes using top down design with meaningful method and class names to abstract away various algorithms in their programs.

The AP CS A (Java) course teaches students how data is stored in a computer as an abstract representation. Students learn about classes where data in variables can be encapsulated and data structures (arrays, ArrayLists, 2D arrays, HashMaps), in general. The AP CS A (Java) course encourages students to make sense of quantities and their relationships in problem situations.

3. **Construct viable arguments and critique the reasoning of others.**
   AP CS A (Java) students are able to analyze situations by breaking them into test cases so they can create programs that solve problems or express their creativity. They justify their conclusions and decisions for choices that they make in their programs, communicate them to others, and respond to the arguments of others about advantages and disadvantages for developing their programs in a certain way. Within all of this, students learn the ethical considerations of sharing their code with others and of finding solutions to CodeHS exercises online.

4. **Model with mathematics.**
   AP CS A students solve many classical mathematics problems within their programs including: making a calculator; computing areas and perimeters of common geometric shapes; adding fractions; generating factorials; solving divisibility problems; calculating and returning doubles, squares, averages, maximums, minimums; unit conversions; determining common combinations and probabilities; working with the unit circle; converting between different number base systems, and more.

5. **Use appropriate tools strategically.**
   Students in the AP CS A (Java) course might use a calculator, a spreadsheet, and other tools to test that their output for mathematical computations is correct so that they eliminate any logical errors in their programs. Students detect possible errors in their code by strategically using estimation and other mathematical knowledge, and by performing code traces and utilizing various debugging strategies.

6. **Attend to precision.**
   The AP CS A (Java) course teaches students how to make choices about how data elements are organized and where data is stored, in general. Students can consider these choices in terms of speed, reliability, accessibility, privacy, and integrity and how those choices impact their programs. Students consider how different primitive types impact what is stored during the execution of their programs. Java is a very specifically typed language meaning that the variable types must be precise. Casting is considered when needing to convert between different types such as int and doubles.

Storing and transforming data are all taught as part of the AP CS A (Java) course. Students learn to use various types of data structures to store data, as well as how to select the proper data structure to model a problem. Students have the option to write programs that manipulate data, find elements in data structures, and remove elements from data structures.

7. **Look for and make use of structure.**
   In AP CS A (Java) students learn that the combination of hardware, software, and their programming practices control access to data and systems within the networks they work in.

   General algorithms in programs create structures that can be used repeatedly. Algorithms and programming are the central focus of the AP CS A (Java) course. Students learn the core principles of developing their own algorithms and implementing them in the Java programming language. Algorithms, variables, control, modularity, and program development are all taught in this course.

   Students also make use of structure when working with various data structures like arrays, arraylists, lists, and hashmaps.

8. **Look for and express regularity in repeated reasoning.**
   In the AP CS A (Java) course, students notice if calculations are repeated, and look both for general methods and shortcuts to make make their code more concise and reusable.

Standards for Mathematical Content

AP Java (Mocha)
 The CodeHS AP Java course is a year-long course designed to help students master the basics of Java and equip them to successfully pass the College Board AP Computer Science A Exam at the end of the school year.
Standards for Mathematical Content Addressed
6.RP.1 Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities. **6.RP.3** Use ratio and rate reasoning to solve real-world and mathematical problems, e.g., by reasoning about tables of equivalent ratios, tape diagrams, double number line diagrams, or equations. **6.NS.1** Interpret and compute quotients of fractions, and solve word problems involving

division of fractions by fractions, e.g., by using visual fraction models and equations to represent the problem.
6.NS.2 Fluently divide multi-digit numbers using the standard algorithm.
6.NS.3 Fluently add, subtract, multiply, and divide multi-digit decimals using the standard algorithm for each operation.
6.NS.5 Understand that positive and negative numbers are used together to describe quantities having opposite directions or values (e.g., temperature above/below zero, elevation above/below sea level, credits/debits, positive/negative electric charge); use positive and negative numbers to represent quantities in real-world contexts, explaining the meaning of 0 in each situation.
6.NS.6c Find and position integers and other rational numbers on a horizontal or vertical number line diagram; find and position pairs of integers and other rational numbers on a coordinate plane.
6.NS.7 Understand ordering and absolute value of rational numbers.
6.NS.8 Solve real-world and mathematical problems by graphing points in all four quadrants of the coordinate plane. Include use of coordinates and absolute value to find distances between points with the same first coordinate or the same second coordinate.
6.EE.1 Write and evaluate numerical expressions involving whole-number exponents.
6.EE.2 Write, read, and evaluate expressions in which letters stand for numbers.
6.EE.2c Evaluate expressions at specific values of their variables. Include expressions that arise from formulas used in real-world problems. Perform arithmetic operations, including those involving whole- number exponents, in the conventional order when there are no parentheses to specify a particular order (Order of Operations).
6.EE.3 Apply the properties of operations to generate equivalent expressions.
6.EE. 5 Understand solving an equation or inequality as a process of answering a question: which values from a specified set, if any, make the equation or inequality true? Use substitution to determine whether a given number in a specified set makes an equation or inequality true.
6.EE.6 Use variables to represent numbers and write expressions when solving a real-world or mathematical problem; understand that a variable can represent an unknown number, or, depending on the purpose at hand, any number in a specified set.
6.EE.7 Solve real-world and mathematical problems by writing and solving equations of the form $x + p = q$ and $px = q$ for cases in which p, q and x are all nonnegative rational numbers.
6.EE.8 Write an inequality of the form $x > c$ or $x < c$ to represent a constraint or condition in a real-world or mathematical problem. Recognize that inequalities of the form $x > c$ or $x < c$ have infinitely many solutions; represent solutions of such inequalities on number line diagrams.
6.G.1 Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
6.G.3 Draw polygons in the coordinate plane given coordinates for the vertices; use coordinates to find the length of a side joining points with the same first coordinate or the same second coordinate. Apply these techniques in the context of solving real-world and mathematical problems.
7RP.2 Recognize and represent proportional relationships between quantities.
7.RP.3 Use proportional relationships to solve multistep ratio and percent problems.
7.NS.1 Apply and extend previous understandings of addition and subtraction to add and

subtract rational numbers; represent addition and subtraction on a horizontal or vertical number line diagram.
7.NS.1b Understand p + q as the number located a distance |q| from p, in the positive or negative direction depending on whether q is positive or negative. Show that a number and its opposite have a sum of 0 (are additive inverses). Interpret sums of rational numbers by describing real-world contexts.
7.NS.1d Apply properties of operations as strategies to add and subtract rational numbers.

7.NS.2 Apply and extend previous understandings of multiplication and division and of fractions to multiply and divide rational numbers.
7.NS.3 Solve real-world and mathematical problems involving the four operations with rational numbers.
7.EE.3 Solve multi-step real-life and mathematical problems posed with positive and negative rational numbers in any form (whole numbers, fractions, and decimals), using tools strategically. Apply properties of operations to calculate with numbers in any form; convert between forms as appropriate; and assess the reasonableness of answers using mental computation and estimation strategies.
7.EE.4 Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.
7.G.4 Know the formulas for the area and circumference of a circle and use them to solve problems; give an informal derivation of the relationship between the circumference and area of a circle.
7.G.6 Solve real-world and mathematical problems involving area, volume and surface area of two- and three-dimensional objects composed of triangles, quadrilaterals, polygons, cubes, and right prisms.
7.SP.5 Understand that the probability of a chance event is a number between 0 and 1 that expresses the likelihood of the event occurring. Larger numbers indicate greater likelihood. A probability near 0 indicates an unlikely event, a probability around 1/2 indicates an event that is neither unlikely nor likely, and a probability near 1 indicates a likely event.
8.EE.1 Know and apply the properties of integer exponents to generate equivalent numerical expressions.
8.F.1 Understand that a function is a rule that assigns to each input exactly one output. The graph of a function is the set of ordered pairs consisting of an input and the corresponding output.
N-Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs and data displays.
N-Q.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities.
A-SSE.1 Interpret expressions that represent a quantity in terms of its context.
A-SSE.2 Use the structure of an expression to identify ways to rewrite it.
F-IF.1 Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then f(x) denotes the output of f corresponding to the input x. The graph of f is the graph of the equation y = f(x).
F-BF.1 Write a function that describes a relationship between two quantities.

F-BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context.
F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context.
F-TF.1 Understand radian measure of an angle as the length of the arc on the unit circle subtended by the angle.