# CodeHS

## AP Computer Science Principles
## Common Core State Standards for Math
## Standards Alignment Overview

The AP CS Principles course is an introductory college-level computer science course introducing the basics of programming with Karel the Dog and JavaScript, the basics of designing a web page, how information and images are represented with computers, the functionality and impact of the Internet, and how to analyze and interpret data. Important Note: The AP CS Principles 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.**
   In the CodeHS AP CS Principles course, students continually design and test the websites that they build, and these sites will invariably have bugs, performance issues, browser rendering differences, deprecated code, and more. They persevere in solving them by using different browsers, checking their HTML and CSS code for errors, and looking up specific issues on the Internet or consulting others, as needed until their sites are usable and presentable.

   Errors are opportunities to learn from mistakes. Students also continually test programs that that they build using Karel the Dog and JavaScript, and these programs will invariably have bugs, as well. 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 checking their JavaScript by using the step debugger and other debugging strategies.

   Computing Systems is a core concept throughout the AP CS Principles course. Students learn about various computing devices and how humans interact with them, including devices that extend the capabilities of humans. Students learn about computer organization including the relationship between hardware and software. Troubleshooting computing systems is a core concept of the AP CS Principles 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.**
   The AP CS Principles course encourages students to make sense of quantities and their relationships in problem situations. Students need to consider layouts in design using HTML and CSS, which means that every pixel counts and any shift can throw off the entire look and feel and usability.  When designing user interfaces, students need to think abstractly and conceptually about their designs, and drill down to details and consider the same designs quantitatively within the site code.

   Nearly every computer science course deals with various levels of abstraction, and the AP CS Principles course is no exception. Students create their own JavaScript functions using top down design with meaningful function names to abstract away various algorithms in their programs. The AP CS Principles course encourages students to make sense of quantities and their relationships in problem situations. Students need to consider design layouts with many quantitative decisions in the graphics, animation, and game exercises.

   Algorithms and programming are central to the AP CS Principles course. Students learn the core principles of developing their own algorithms and implementing them in the JavaScript programming language. Algorithms, variables, control, modularity, and program development are all taught in this course.

   The AP CS Principles course teaches students how data is stored in a computer as an abstract representation. Students learn exactly how text and image data is organized and stored as physical bits in a computing system. Students also learn how sensors must convert physical data into a digital representation that can be stored in a computer, and how data collection can be automated with sensors.

3. **Construct viable arguments and critique the reasoning of others.**
   AP CS Principles students are able to analyze situations by breaking them into cases so they can create websites that are user-friendly for all. They justify their conclusions and decisions for choices that they make on their sites, communicate them to others, and respond to the arguments of others about advantages and disadvantages for designing and developing their site in a certain way.

   Computing has had significant impacts in several fields, and students critique the reasoning of others when discussing these various topics. In this course, students learn about the positive and negative impacts the Internet has had on culture, social interactions, safety, and privacy and the impacts computing has on animation, gaming, and digital art. Students learn the ethical considerations of sharing their code with others, and finding solutions to CodeHS exercises online. Students learn about the importance of cybersecurity and the various security measures we take to protect information and privacy on the Internet.

AP CS Principles students are able to analyze situations by breaking them into 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.

4. **Model with mathematics.**
   Students in the CodeHS AP CS Principles course use geometry to solve website design layout problems. Students in the course can also apply what they know and are comfortable making assumptions and approximations to simplify a complicated designs, realizing that these may need revision later.

   Students in the CodeHS AP CS Principles course use geometry to solve graphics, animation and game design layout problems. Students in the course can also apply what they know and are comfortable making assumptions and approximations to simplify a complicated designs, realizing that these may need revision later. Finally, students solve many classical mathematics problems (e.g., making a calculator, generating the Fibonacci sequence, creating sets) with their programs.

5. **Use appropriate tools strategically.**
   Students might use pencil and paper, concrete models, a ruler, a calculator, a spreadsheet, and other tools to create their sites and apps in the AP CS Principles course. Students detect possible errors in their code by strategically using estimation and other mathematical knowledge.

   Students might use pencil and paper, concrete models, a ruler, a calculator, and other tools to create their programs in the AP CS Principles course. Students detect possible errors in their code by strategically using estimation and other mathematical knowledge.

6. **Attend to precision.**
   The AP CS Principles course teaches students how to make choices about how data elements are organized and where data is stored on websites. Students can consider these choices in terms of speed, reliability, accessibility, privacy, and integrity and how those choices impact their own site development.

   Storing, transforming, and visualizing data are all taught as part of the AP CS Principles 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, remove elements from data structures, and even visualize music data in the Supplemental Units of the course.

7. **Look for and make use of structure.**
Students look closely to discern a pattern or structure in websites and applications. Complex websites are designed as systems of interacting or nested modules, each with a specific role, coordinating for a common overall purpose. These modules can be combinations of data (images, text, etc.) which allow for better management of complex sites. In the Internet unit, students learn learn about network communication and organization, basic Internet protocols, and Internet addressing, which all also make use of structural elements.

   General algorithms in programs create structures that can be used repeatedly. Algorithms and programming are the central focus of the AP CS Principles course. Students learn the core principles of developing their own algorithms and implementing them in the JavaScript 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 number systems to understand and use digital information. They encode text with binary and work with hexadecimals in pixel images. Students then use the digital information to manipulate images with consideration for data and lossy compression.

   Finally, in the AP CS Principles course, students make use of structures in data when they collect, visualize, and interpret data. They are encouraged to be systematic when collecting data and to look for trends that can inform their data-driven insights.

8. **Look for and express regularity in repeated reasoning.**
In the AP CS Principles 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 CS Principles** |
|---|
| AP Computer Science Principles is the newest AP® course from the College Board. This course introduces students to the foundational concepts of computer science and explores the impact computing and technology have on our society. Students learn about the internet, digital information, programming, data, and apply these concepts through creative projects, while building their portfolio. |

**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.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.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.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.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.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.
**6.SP.2** Understand that a set of data collected to answer a statistical question has a distribution which can be described by its center, spread, and overall shape.
**6.SP.5** Summarize numerical data sets in relation to their context.
**7RP.2** Recognize and represent proportional relationships between quantities.
**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.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.SP.1** Understand that statistics can be used to gain information about a population by examining a sample of the population; generalizations about a population from a sample are valid only if the sample is representative of that population. Understand that random sampling tends to produce representative samples and support valid inferences.

**7.SP.4** Use measures of center and measures of variability for numerical data from random samples to draw informal comparative inferences about two populations.

**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.

**8.SP.1** Construct and interpret scatter plots for bivariate measurement data to investigate patterns of association between two quantities. Describe patterns such as clustering, outliers, positive or negative association, linear association, and nonlinear association.

**8.SP.2** Know that straight lines are widely used to model relationships between two quantitative variables. For scatter plots that suggest a linear association, informally fit a straight line, and informally assess the model fit by judging the closeness of the data points to the line.

**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.

**A-SSE.1** Interpret expressions that represent a quantity in terms of its context.

**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.

**S-ID.3** Interpret differences in shape, center, and spread in the context of the data sets, accounting for possible effects of extreme data points (outliers).

**S-ID.9** Distinguish between correlation and causation.
**S-IC.1** Understand statistics as a process for making inferences about population parameters based on a random sample from that population.
**S-IC.3** Recognize the purposes of and differences among sample surveys, experiments, and observational studies; explain how randomization relates to each.
**S-IC.6** Evaluate reports based on data.