



CodeHS

AP Computer Science Principles Course Overview

Introduction

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.

With a unique focus on creative problem solving and real-world applications, the CodeHS AP Computer Science Principles course gives students the opportunity to explore several important topics of computing using their own ideas and creativity, use the power of computing to create artifacts of personal value, and develop an interest in computer science that will foster further endeavors in the field.

Course Overview

Prerequisites: There are no official prerequisites for the CodeHS AP Computer Science Principles course. This course is meant to be a first time introduction to computer science, and does not require students to come in with any computer programming experience. However, we recommend that students take our Introduction to Computer Science prior to our AP courses (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 AP courses.

Learning Environment: The course utilizes a blended classroom approach. The content is a mix of web-based and physical activities. Students will write and run code in the browser, create websites and digital presentations, and engage in in-person collaborative exercises with classmates. 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, written programming exercises, free response exercises, collaborative creation projects and research projects.

Programming Environment: Students write and run programs in the browser using the CodeHS editor. Students will be able to write both text based and block based JavaScript programs, and students will use Processing.js to create graphical programs. They will also create webpages using HTML, CSS, and JavaScript. These webpages will be hosted on the CodeHS website so

that they can keep a running portfolio of their creative projects, and easily share their programs with the world. Students gain programming experience early on in the course that will enable them to explore the rest of the course topics through computational thinking practices.

Quizzes: At the end of each unit, students take a summative multiple choice unit quiz in the style of the AP Exam that assesses their knowledge of the concepts covered in the unit. Included in each lesson is a formative short quiz of various question types, including multiple choice, free response, and matching. The course also provides an AP Test Practice unit with a cumulative AP Practice Multiple Choice Test.

Course Objectives

This course is based directly off of the College Board AP Computer Science Principles Curriculum Framework. We recommend reading the curriculum framework [here](#) for context. The main course objectives are summarized below in the six computational thinking practices and seven big ideas for the course.

Computational Thinking Practices:

The six computational thinking practices represent important aspects of the work that computer scientists engage in, and are denoted here by P1 through P6:

- P1: Connecting Computing
- P2: Creating Computational Artifacts
- P3: Abstracting
- P4: Analyzing Problems and Artifacts
- P5: Communicating
- P6: Collaborating

Big Ideas:

The seven big ideas of the course encompass foundational ideas of the field of computer science, and are denoted here by B1 through B7:

- B1: Creativity
- B2: Abstraction
- B3: Data and Information
- B4: Algorithms
- B5: Programming
- B6: The Internet
- B7: Global Impact

The AP Through Course Assessment:

The through course assessment is a set of performance tasks designed to gather evidence of student proficiency in the learning objectives. The AP Performance Tasks (PTs) are in-class assessments, administered by the teacher, that allow students to exemplify their learning through

authentic, “real-world” creations. For more information about the AP Performance Tasks, refer to the [curriculum framework](#).

The two performance tasks as defined by College Board are:

1. Explore - Implications of Computing Inventions
 - Students explore the impacts of computing on social, economic, and cultural areas of our lives.
2. Create - Applications from Ideas
 - Students create computational artifacts through the design and development of programs.

Students will gain the experience necessary to complete the PTs in class. Each unit comes with practice PTs in which students will research topics in computing, and create their own digital artifacts. Students will create and maintain a website that will hold each student creation throughout the course. This will serve as a running portfolio of each creative project the student completes. Sufficient time is set aside in the course for students to prepare for and complete both PTs.

The AP Exam:

This course will prepare students for the multiple choice AP Computer Science Principles examination. Each lesson comes with quizzes to test essential knowledge for the AP Exam. Each unit includes a cumulative AP style multiple choice exam to test understanding of the concepts in the unit, and provide immediate feedback to the student.

Course Breakdown

High level overview of each unit in the course:

Unit 1: Web Development (3 weeks)

In this unit, students will go through a high level introduction to HTML, CSS, and the processes involved in viewing web pages on the internet. Students will create several simple web pages using the CodeHS online editor to gain practice using the various features of HTML and CSS. This unit culminates with each student making their own website about themselves, hosted on their own custom CodeHS URL. This personal website will be continually improved by the student as they continue on in the course, and will serve as a running portfolio of each creative project they create in the course.

Unit 2: Introduction to Programming with Karel the Dog (4 weeks)

This course begins with a strong focus on programming in order to allow students to create computational artifacts early on in the course. Students will be able to use their knowledge of programming to explore the future topics in the course.

We use Karel, a dog that only knows how to move, turn left, and place tennis balls in his world, to show students what it means to program, and allow students to focus on computational problem solving. Students will learn about the need for programming languages, the uses of programs, how to write programs to solve computational problems, how to design algorithms, how to analyze and compare potential solutions to programming problems, and learn the value and challenges involved in collaborating with others to solve programming problems. Students will use the grid coloring functionality of Karel to create a digital painting, and embed this program in their portfolio website.

Unit 3: Basic JavaScript (4 weeks)

This unit introduces students to the basics of JavaScript, and gives students practice writing JavaScript programs to solve general problems. Students will be able to compare and contrast JavaScript with Karel and identify the abstractions Karel provides over JavaScript. Other topics in this unit include data structures, APIs, the importance of programming style, and the impact programming has had on the types of problems that can be solved. Students will use their knowledge of JavaScript to write a graphical program that tells a story, and embed this program in their portfolio website.

Unit 4: Digital Information (6 weeks)

In this unit, students will learn about the various ways we represent information digitally. Topics covered include number systems, encoding data, programmatically creating pixel images, comparing data encodings, compressing and encrypting data. Students will work in pairs to develop their own data encryption algorithms, and attempt to crack the encryptions of their peers. Their text encryption tool will be embedded in their portfolio websites.

Unit 5: The Internet (6 weeks)

This unit explores the structure and design of the internet, and how this design affects the reliability of network communication, the security of data, and personal privacy. Students will learn about the protocols and algorithms used in the internet, and the importance of cybersecurity. Students will prepare a short presentation discussing one concrete way the internet has impacted society, deliver a presentation to their classmates, and embed this presentation in their portfolio website.

Unit 6: Data (5 weeks)

In this unit, students will explore using computational tools to store massive amounts of data, manipulate and visualize data, find patterns in data, and pull conclusions from data. Students will consider how the modern wealth of data collection has impacted society in positive and negative ways. Students will work in teams to investigate a question of personal interest, and use public data to present a data driven insight to their peers. They will embed this presentation and any visualizations in their portfolio websites.

Unit 7: Performance Tasks (7 weeks)

This time is set aside for students to prepare for and create their AP Performance Tasks. Students will be given the chance to review course content and practice the skills necessary to complete each performance task. The Explore PT will be administered over 8 hours of class time, and the Create PT will be administered over 12 hours of class time.

Unit 8: Review for the AP Exam (2 weeks)

This unit gives students a review of the topics covered in the course and provides practice solving AP Exam style multiple choice questions.

Unit 9: Final Project (Remainder of the school year)

In this unit students will brainstorm their own final project, discuss their ideas with their peers, scope their project to fit within the time constraints of the class, plan out milestones for incremental development, and create their own final product from scratch. This project allows students to think creatively about the applications of the concepts covered in the course, and create something of personal value.