

Plainfield School District 202 – AP Computer Science Classes

AP Computer Science Principles

Open to: 10 – 12th Grade

Prerequisites: Algebra 1

Length of Term: 1 Year

AP Computer Science Principles is a very interesting course to teach. Its focus is more on Computer Science and less on programming. There is a programming component but the course is more about technology and innovation than anything else. The College Board has designed the course around “7 Big Ideas”: Creativity, Abstraction, Data and Information, Algorithms, Programming, The Internet, and Global Impact. This is meant to be an introductory class to computer science but with the rigors and expectations of an AP level course. The College Board will be expecting that every student enrolled in the course completes 3 parts to obtain an AP score. The first responsibility is to write a paper describing a technological innovation and its global impact. Secondly they will have to submit a programming artifact (that can be done in any language) and a video demonstrating its use. The programming artifact must be something they created themselves and can be done with a partner. Lastly, there is an exam that is 75 multiple choice questions in a 2 hour time period. The combined score on the 3 items will determine the AP score they will be given. AP level computer science courses should be given math credit, but not meant to replace the 2 years of courses with an Algebra component and 1 year with a geometry component. APCS Principles is accepted by a relatively small but growing number of colleges and universities as an introductory computer science course.

Students to consider for recommendation:

- Students who are ready for the rigor of an AP course but have no prior programming experience
- Students who have an interest in technology, how things work, and the impact it has on society

AP Computer Science A (Current AP offering)

Open to: 10 – 12th Grade

Prerequisites: Programming 2 or Principles

Length of Term: 1 Year

AP Computer Science A has a much different focus than APCS Principles. The focus of APCS A is on software development and uses Java as the language. Students will be interacting with code every day of the class, either writing it themselves, interpreting it, or using it for a specific purpose. This class walks through the fundamentals of program design and ends on higher order and abstract principles of programming, focusing on object-oriented programming. Most students think of programming as writing a specific set of instructions that will be handled in a sequential way. Object-oriented design does not follow this approach. We develop objects that have the ability to store information and perform specific tasks. Program execution is then dictated on the interactions of the objects you introduce in the program, the introduction of different objects together will yield unique executions. Student responsibilities will be the same as most courses but will place an emphasis on intensive lab work followed up with written quizzes to assess understanding. The AP exam consists of 60 multiple choice questions followed by 4 free response questions performed in two separate 90 minute parts. AP level computer science courses should be given math credit, but not meant to replace the 2 years of courses with an Algebra component and 1 year with a geometry component. APCS A is accepted for credit by a large number of colleges and universities as an introductory computer science class.

Students to consider for recommendation:

- Students who are interested in STEM pathway and are ready for the rigors of an AP course
- Prior programming experience is nice but not a must have

Computer Science Principles

Curriculum Overview



Why Computer Science? Every 21st century student should have the opportunity to learn computer science. The basics of computer science help nurture creativity and problem-solving skills, and prepare students for a future in any field or career.

Advanced Placement Computer Science for All Students!



Code.org's Computer Science Principles is an introductory Advanced Placement (AP[®]) course designed to broaden participation in computer science. Code.org is recognized by the College Board as an endorsed provider of AP[®] Computer Science Principles curriculum and professional development. The course has been reviewed by the College Board and is pre-approved to pass the course professional development is held to (and goes beyond) the standards of AP[®] Summer Institutes. The official AP[®] exam launched in the 2016-17 school year.

Engaging Curriculum

Our team designed the AP[®] Computer Science Principles curriculum to support students and teachers new to the discipline. The curriculum includes daily lesson plans made up of inquiry-based activities, videos, assessments, and computing tools, allowing teachers to guide and learn alongside students as they discover core computing concepts.



One-Year Professional Learning Program

Summer: Teachers attend a 5-day in-person workshop designed to introduce the CS concepts from the curriculum, AP[®] elements of the course, and core teaching practices. (Travel may be required.)

School Year: Teachers continue with job-embedded workshops and online modules focused on supporting their first year of implementation.

Hundreds of teachers have participated. The majority say "It's the best professional development I've ever attended."



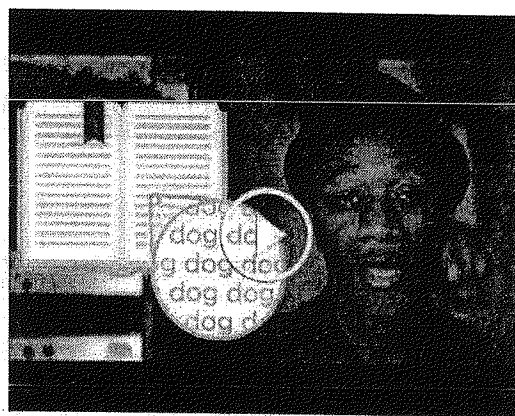
"Hands-down the best PD I've ever been to. This week modeled the equity that we will bring to our computer science principles classrooms."



"Pretty much every student wants to take the next computer science courses we'll offer."

Curriculum Features:

- Daily instructional lesson plans that include inquiry/equity-based pedagogy and background content
- Formative and summative assessments, project exemplars and rubrics
- Widgets and simulators for exploring computing concepts like text compression and the Internet
- Concept and tutorial videos for students, and teaching tips-and-tricks videos for teachers
- Code Studio—a learning platform where students interact with lesson materials and tools, and where teachers access a dashboard to see student work and progress
- App Lab—a JavaScript programming environment in Code Studio, designed for creating event-driven web apps with block-to-text workspace and debugging capabilities



Aloe Blacc explains DIGITAL COMPRESSION

Unit Overview

Unit 1: The Internet	Learn how the multi-layered systems of the Internet function as you collaboratively solve problems and puzzles about encoding and transmitting data, both 'unplugged' and using Code.org's Internet Simulator.
Unit 2: Digital Information	Use a variety of digital tools to look at, generate, clean, and manipulate data to explore the relationship between information and data. Create and use visualizations to identify patterns and trends.
Unit 3: Algorithms and Programming	Learn the JavaScript language with turtle programming in Code.org's App Lab. Learn general principles of algorithms and program design that are applicable to any programming language.
Unit 4: Big Data and Privacy	Research current events around the complex questions related to public policy, law, ethics, and societal impact. Learn the basics of how and why modern encryption works.
Unit 5: Building Apps	Continue learning how to program in the JavaScript language. Use Code.org's App Lab environment to create a series of applications that live on the web. Each app highlights a core concept of programming.
Unit 6: Performance Tasks	Design a project plan, then work on and complete your AP [®] Performance Task projects for submission to the College Board.

Apply now for professional learning!

<http://code.org/csp/pd>

For curriculum, videos, and support documents, visit:

<http://code.org/csp>

Code.org is a 501(c)3 non-profit dedicated to expanding participation in computer science education by making it available in more schools, and increasing participation by women and underrepresented students of color. The Code.org vision is that every student in every school should have the opportunity to learn computer programming.

