AP® COMPUTER SCIENCE PRINCIPLES

About the Advanced Placement Program® (AP®)

The Advanced Placement Program® has enabled millions of students to take college-level courses and earn college credit, advanced placement, or both, while still in high school. AP Exams are given each year in May. Students who earn a qualifying score on an AP Exam are typically eligible, in college, to receive credit, placement into advanced courses, or both. Every aspect of AP course and exam development is the result of collaboration between AP teachers and college faculty. They work together to develop AP courses and exams, set scoring standards, and score the exams. College faculty review every AP teacher's course syllabus.

AP Computer Science Program

There are two computer science offerings, and students can take either course in any order. The AP Computer Science A course and exam continues to focus on computing skills related to programming in Java. The new AP Computer Science Principles course complements AP Computer Science A as it aims to broaden participation in the study of computer science. The courses underscore the importance of communicating solutions appropriately and in ways that are relevant to current societal needs. AP Computer Science courses can help address traditional issues of equity, access, and broadening participation in computing while providing a strong and engaging introduction to fundamental areas of the discipline.

AP Computer Science Principles Course Overview

The AP Computer Science Principles course is designed to be equivalent to a first- semester introductory college computing course. In this course, students will develop computational thinking skills vital for success across all disciplines, such as using computational tools to analyze and study data and working with large data sets to analyze, visualize, and draw conclusions from trends. The course engages students in the creative aspects of the field by allowing them to develop computational artifacts based on their interests. Students will also develop effective communication and collaboration skills by working individually and collaboratively to solve problems, and will discuss and write about the impacts these solutions could have on their community, society, and the world.

PREREQUISITES

It is recommended that a student in the AP Computer Science Principles course should have successfully completed a first year high school algebra course with a strong foundation on basic linear functions and composition of functions, and problem solving strategies that require multiple approaches and collaborative efforts. In addition, students should be able to use a Cartesian (x, y) coordinate system to represents points in a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical and computational reasoning that will be applied throughout the study of the course.

Computer Language

Unlike AP Computer Science A, which is taught in Java, the AP Computer Science Principles course does not have a designated programming language. Teachers select the programming language(s) that is most appropriate for their students.

AP Computer Science Principles Course Content

The following are the major areas of study, or big ideas, that are foundational to studying computer science:

- Creativity: Computing is a creative activity. Creativity and computing are prominent forces in innovation; the innovations enabled by computing have had and will continue to have far-reaching impact.
- Abstraction: Abstraction reduces information and detail to facilitate focus on relevant concepts. It is a process, a strategy, and the result of reducing detail to focus on concepts relevant to understanding and solving problems.
- Data and Information: Data and information facilitate the creation of knowledge. Computing enables and empowers new methods of information processing, driving monumental change across many disciplines — from art to business to science.
- Algorithms: Algorithms are used to develop and express solutions to computational problems. Algorithms realized in software have affected the world in profound and lasting ways.
- Programming: Programming enables problem solving, human expression, and creation of knowledge. Programming and the creation of software has changed our lives. It results in the creation of software, and facilitates the creation of computational artifacts, such as music, images, and visualizations.
- The Internet: The Internet pervades modern computing. The Internet and the systems built on it have had a profound impact on society. Computer networks support communication and collaboration.
- Global Impact: Computing has global impact. Our methods for communicating, collaborating, problem solving, and doing business have changed and are changing due to innovations enabled by computing.

Computational Thinking Practices

The course also incorporates computational thinking practices that set clear expectations of what students will do in the course:

- Connecting Computing – Students learn to draw connections between different computing concepts.
- Creating computational artifacts – Students engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems.
- Abstracting – Students use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity.
- Analyzing problems and artifacts – Students design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work others have produced.
- Communicating – Students describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of those artifacts.
- Collaborating – Students collaborate on a number of activities, including investigation of questions using data sets and in the production of computational artifacts.

AP Computer Science Principles Exam Structure

AP COMPUTER SCIENCE PRINCIPLES EXAM: 2 HOURS

Assessment Overview

This assessment comprises two parts: performance tasks and the end-of-course AP Exam and the through-course AP assessment.

The AP Computer Science Principles Exam will be a multiple-choice, paper and pencil exam.

The two performance tasks require students to explore the impacts of computing and create computational artifacts through programming.

Format of Assessment

AP COMPUTER SCIENCE PRINCIPLES EXAM: 2 HOURS (60% of AP Exam score)

- Multiple Choice (single- and multiple-select) | 74 Questions | 120 minutes | 60% of assessment score

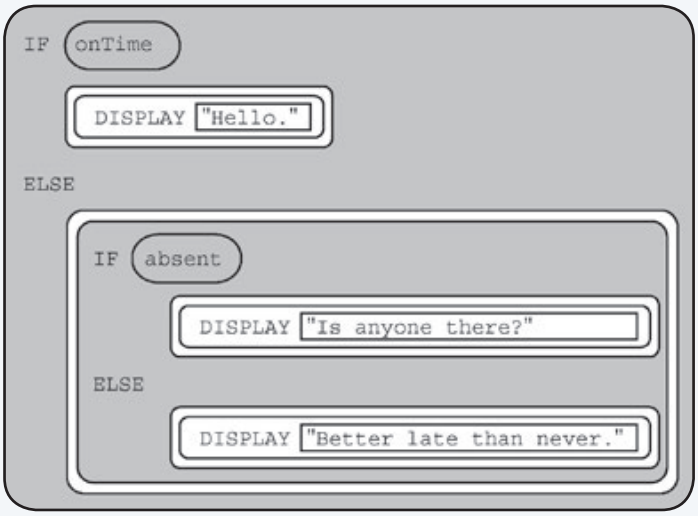
AP COMPUTER SCIENCE PRINCIPLES THROUGH-COURSE PERFORMANCE TASKS (2) (Combined 40% of AP Exam Score):

- Explore – Impact of Computing Innovations | 8 hours (classroom time) | 16% of assessment score
- Create – Application to Ideas |12 hours (classroom time) | 24% of assessment score

AP COMPUTER SCIENCE PRINCIPLES SAMPLE EXAM QUESTIONS

Sample Multiple-Choice Question

Consider the code segment below.



If the variables `onTime` and `absent` both have the value `false`, what is displayed as a result of running the code segment?

(A) `Is anyone there?`

(B) `Better late than never.`

(C) `Hello. Is anyone there?`

(D) `Hello. Better late than never.`

Answer: B

Performance Task: Create – Applications from Ideas

- This performance task focuses on students developing computer programs and describing significant aspects of the program that allow it to run correctly.
  - Students have the flexibility to write programs that reflects their interests (e.g., their desire to solve a problem; program a game; or produce digital art appealing to a specific audience, etc.) This allows students to engage in the study of computer science from a creative perspective. Students will provide evidence of their knowledge of important programming concepts such as developing algorithms and using abstractions. Students are required to submit an individual program but are able to collaborate on the development of their program.

Performance Task: Explore – Impacts of Computing Innovations

- This performance task focuses on students using and applying computational analysis in the exploration of a significant computing innovation to determine and describe the impact of the innovation on people and society.
  - Students select and explore an innovation of their choosing. Then, they create a computational artifact about the innovation and describe how it works and how it used, its purpose, how it consumes and/or produces data, and the harmful and beneficial effects of the innovation on people and society.

Educators: apcentral.collegeboard.org/apcsp

Students: apstudent.collegeboard.org/apcsp

00678-0009 (Updated June 2017)