

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 or concurrently:

- AP Computer Science A focuses on computing skills related to programming in Java.
- AP Computer Science Principles provides students with a broad introduction to computer science and how it relates to other fields.

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

AP Computer Science Principles introduces students to the breadth of the field of computer science. In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts, and contribute to a computing culture that is collaborative and ethical.

PREREQUISITES

It is recommended that students in the AP Computer Science Principles course have successfully completed a first-year high school algebra course with a strong foundation of basic linear functions, 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 represent points on a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.

Prior computer science experience is not required to take this course.

COMPUTER LANGUAGE

AP Computer Science Principles does not have a designated programming language. Teachers have the flexibility to choose a programming language(s) that is most appropriate for their students to use in the classroom.

AP Computer Science Principles Course Content

The following are the major areas of study, or big ideas, that serve as the foundation of the course, enabling students to create meaningful connections among concepts and develop deeper conceptual understanding:

- **Creative Development:** When developing computing innovations, developers can use a formal, iterative design process or a less rigid process of experimentation, and will encounter phases of investigating and reflecting, designing, prototyping, and testing. Collaboration is an important tool at any phase of development.
- **Data:** Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge.
- **Algorithms and Programming:** Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems.
- **Computing Systems and Networks:** Computer systems and networks are used to transfer data.
- **Impact of Computing:** Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues.

Each big idea is broken down into teachable segments called topics.

AP Computer Science Principles Computational Thinking Practices

The following computational thinking practices describe what skills students should develop during the course:

- **Computational Solution Design:** Design and evaluate computational solutions for a purpose.
- **Algorithms and Program Development:** Develop and implement algorithms.
- **Abstraction in Program Development:** Develop programs that incorporate abstractions.
- **Code Analysis:** Evaluate and test algorithms and programs.
- **Computing Innovations:** Investigate computing innovations.
- **Responsible Computing:** Contribute to an inclusive, safe, collaborative, and ethical computing culture.

AP Computer Science Principles End-of-Course Exam Structure

AP COMPUTER SCIENCE PRINCIPLES END-OF-COURSE
EXAM: 2 HOURS

Assessment Overview

The AP Computer Science Principles Exam assesses student understanding of the computational thinking practices and learning objectives outlined in the course framework. The exam consists of the Create performance task and an end-of-course AP Exam. The Create performance task requires at least 12 hours of dedicated class time for students to complete. The end-of-course exam is 2 hours long and includes 70 multiple-choice questions.

Format of Assessment

Section I: Multiple-choice | 70 Questions | 2 Hours |
70% of Exam Score

- 57 single-select multiple-choice questions.
- 5 single-select multiple-choice questions with reading passage about a computing innovation.
- 8 multi-select multiple-choice questions.

Section II: Create Performance Task | 1 Question |
At least 12 hours of class | 30% of Exam Score

- Through-course performance task
- Assesses Computational Thinking Practices 1, 2, 3, and 4

Exam Components

Sample Multiple-Choice Questions

A digital photo file contains data representing the level of red, green, and blue for each pixel in the photo. The file also contains metadata that describe the date and geographic location where the photo was taken. For which of the following goals would analyzing the metadata be more appropriate than analyzing the data?

- (A) Determining the likelihood that the photo is a picture of the sky
- (B) Determining the likelihood that the photo was taken at a particular public event
- (C) Determining the number of people that appear in the photo
- (D) Determining the usability of the photo for projection onto a particular color background

A certain computer has two identical processors that are able to run in parallel. Each processor can run only one process at a time, and each process must be executed on a single processor. The following table indicates the amount of time it takes to execute each of three processes on a single processor. Assume that none of the processes are dependent on any of the other processes.

Process	Execution Time on Either Processor
X	60 seconds
Y	30 seconds
Z	50 seconds

Which of the following best approximates the minimum possible time to execute all three processes when the two processors are run in parallel?

- (A) 60 seconds
- (B) 70 seconds
- (C) 80 seconds
- (D) 90 seconds

Create Performance Task

In the Create Performance Task, each student designs and implements a computer program that might solve a problem, enable innovation, explore personal interests, or express creativity. Their final program code is accompanied by a video that displays the running of their program and demonstrates functionality, as well as written responses to task prompts.

- ◆ Students have the flexibility to write programs that reflect their interests, which allows them 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 may choose to collaborate during the development of their program, but must submit independently created videos of the program running. Students will independently respond to prompts to demonstrate their understanding of their program, the data, algorithms and abstractions the program uses, and how it will behave under different circumstances. Students may develop their program using a program language of their choice. The programming language selected should contain functionality that is specified in the performance task. HTML is not an acceptable programming language for the Create Performance Task.
- ◆ The Student Handouts for the Create Performance Task can be accessed [here](#).