AP® Computer Science A

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. AP Computer Science A, although more oriented toward programming, complements AP Computer Science Principles, which focuses on the innovative aspects of computing and computational thinking. Students can take the courses in any order.

AP Computer Science A focuses on object-oriented programming and problem-solving using the Java programming language.

AP Computer Science Principles curriculum is built around fundamentals of computing. Students engage with the course content by developing computational artifacts and analyzing data, information, or knowledge represented for computational use.

AP Computer Science A Course Overview

AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.

COLLEGE COURSE EQUIVALENT

AP Computer Science A is equivalent to an introductory college course in computer science.

PREREQUISITES

It is recommended that a student in the AP Computer Science A course has 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. AP Computer Science A builds upon a foundation of mathematical reasoning that should be acquired prior to taking this course.

PROGRAMMING LANGUAGE

The AP Computer Science A course requires that solutions of problems be written in the Java programming language. Because the Java programming language is extensive, with far more features than could be covered in a single introductory course, the AP Computer Science A Exam covers a subset of Java.

LAB REQUIREMENT

The AP Computer Science A course must include a minimum of 20 hours of hands-on, structured lab experiences to engage students in individual or group problem-solving. Thus, each AP Computer Science A course includes a substantial lab component in which students design solutions to problems, express their solutions precisely (e.g., in the Java programming language), test their solutions, identify and correct errors (when mistakes occur), and compare possible solutions. College Board has developed several labs that are aligned to the course framework that fulfill the 20-hour lab requirement. The class period recommendations provided in the unit guides account for the time needed to complete each lab activity as described in the lab guide.

AP Computer Science A Course Content

The course content is organized into four commonly taught units:

- **Unit 1:** Using Objects and Methods
- **Unit 2:** Selection and Iteration
- **Unit 3:** Class Creation
- **Unit 4:** Data Collections

AP Computer Science A Computational Thinking Practices

The computational thinking practices and skills for AP Computer Science A describe what a student should be able to do while exploring course concepts:

- **Design Code:** Determine an appropriate program design and develop algorithms.
- **Develop Code:** Write and implement program code.
- **Analyze Code:** Determine the output or result of given program code or explain why code may not work as intended.
- **Document Code and Computing Systems:** Describe the behavior and conditions that produce specified results in a program.
- **Use Computers Responsibly:** Understand the ethical and social implications of computer use.

AP Computer Science A Exam Structure

AP COMPUTER SCIENCE A EXAM: 3 HOURS

Assessment Overview

The AP Computer Science A Exam assesses student understanding of the computational thinking practices, learning objectives, and essential knowledge statements outlined in the course framework. The exam is 3 hours long and includes 42 multiple-choice questions and 4 free-response questions. As part of the exam, students will be given the Java Quick Reference, which lists accessible methods from the Java library that may be included on the exam.

Format of Assessment

Section I: Multiple-choice | 42 Questions | 90 Minutes | 55% of Exam Score

- Includes mostly discrete questions, with up to two sets of multiple questions (typically two questions per set)
- Assesses Computational Thinking Practices 1, 2, 3, 4, and 5

Section II: Free-response | 4 Questions | 90 Minutes | 45% of Exam Score

- Question 1: Methods and Control Structures (**7 points**)
- Question 2: Class Design (**7 points**)
- Question 3: Data Analysis with `ArrayList` (**5 points**)
- Question 4: 2D Array (**6 points**)

Exam Components

Sample Multiple-Choice Question

Consider the following code segment.

```
String str1 = "LMNOP";
String str2 = str1.substring(3);
str2 += str1.substring(2, 3);
```

What is the value of `str2` after executing this code segment?

(A) `"OPN"`

(B) `"OPNO"`

(C) `"NOPM"`

(D) `"NOPMN"`

Correct Answer: A

Sample Free-Response Question: Class Design

The `CupcakeMachine` class, which you will write, represents a cupcake vending machine, an automated machine that dispenses cupcakes.

`CupcakeMachine` objects are created by calls to a constructor with two parameters.

- The first parameter is an `int` that represents the number of cupcakes that the vending machine has been stocked with. Assume that this value will be greater than or equal to `0`.
- The second parameter is a `double` that represents the cost, in dollars, per cupcake. Assume that this value will be greater than `0.0`.

The `CupcakeMachine` class contains a `takeOrder` method, which determines whether a cupcake order can be filled. A cupcake order can be filled if there are at least as many cupcakes in the vending machine as there are in the order.

A cupcake order is represented by a single `int` parameter to the `takeOrder` method. Assume that all values passed to the `takeOrder` method are positive. If the order cannot be filled because the vending machine does not have enough cupcakes, the `takeOrder` method should return the message `"Order cannot be filled"`. In this case, the number of cupcakes available in the machine is unchanged and no order number is given to the order.

For example, suppose a `CupcakeMachine` named `c1` is constructed with 10 cupcakes and a cost of $1.75 per cupcake. If the first call to `c1` is `c1.takeOrder(2)`, it will return `"Order number 1, cost $3.5"`. If the second call to `c1` is `c1.takeOrder(3)`, it will return `"Order number 2, cost $5.25"`. If the third call to `c1` is `c1.takeOrder(10)`, it will return `"Order cannot be filled"` because the machine does not have enough cupcakes after the first two calls. If the fourth call to `c1` is `c1.takeOrder(1)`, it will return `"Order number 3, cost $1.75"`.

Write the complete `CupcakeMachine` class.

Educators: apcentral.collegeboard.org/courses/ap-computer-science-a

Students: apstudent.collegeboard.org/courses/ap-computer-science-a

CollegeBoard