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, 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 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.

PREREQUISITES

It is recommended that students 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

The course requires that solutions of problems be written in the Java programming language. Because the Java programming language is extensive, 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. Each course includes a substantial lab component in which students design solutions to problems, express their solutions precisely, test their solutions, identify and correct errors, and compare possible solutions. The College Board has developed several labs that are aligned to the course framework that fulfill the 20-hour lab requirement.

AP Computer Science A Course Content

The course content is organized into ten commonly taught units:

- **Unit 1:** Primitive Types
- **Unit 2:** Using Objects
- **Unit 3:** Boolean Expressions and `if` Statements
- **Unit 4:** Iteration
- **Unit 5:** Writing Classes
- **Unit 6:** Array
- **Unit 7:** `ArrayList`
- **Unit 8:** 2D Array
- **Unit 9:** Inheritance
- **Unit 10:** Recursion

The following big ideas serve as the foundation of the course, enabling students to create meaningful connections among concepts:

- **Modularity:** Modularity in object-oriented programming allows us to use abstraction to break complex programs down into individual classes and methods.
- **Variables:** Variables create data abstractions, as they can represent a set of possible values or a group of related values.
- **Control:** Doing things in order, making decisions, and doing the same process multiple times are represented in code by using control structures.
- **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.

AP Computer Science A Computational Thinking Practices

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

- **Program Design and Algorithm Development:** Determine required code segments to produce a given output.
- **Code Logic:** Determine the output, value, or result of given program code given initial values.
- **Code Implementation:** Write and implement program code.
- **Code Testing:** Analyze program code for correctness, equivalence, and errors.
- **Documentation:** Describe the behavior and conditions that produce identified results in a program.

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 and learning objectives outlined in the course framework. The exam is 3 hours long and includes 40 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 in the exam.

Format of Assessment

Section I: Multiple-choice | 40 Questions | 90 Minutes | 50% of Exam Score

- Mostly individual questions, with one or two sets of multiple questions (typically two to three questions per set).
- Computational Thinking Practices 1, 2, 4, and 5 are assessed.

Section II: Free-response | 4 Questions | 90 Minutes | 50% of Exam Score

- Question 1: Methods and Control Structures (**9 points**).
- Question 2: Class (**9 points**).
- Question 3: Array/`ArrayList` (**9 points**).
- Question 4: 2D Array (**9 points**).
- Computational Thinking Practice 3 is assessed.

Exam Components

Sample Multiple-Choice Question

Which of the following statements assigns a random integer between 25 and 60, inclusive, to `rn`?

(A) `int rn = (int) (Math.random() * 25) + 36;`

(B) `int rn = (int) (Math.random() * 25) + 60;`

(C) `int rn = (int) (Math.random() * 26) + 60;`

(D) `int rn = (int) (Math.random() * 36) + 25;`

(E) `int rn = (int) (Math.random() * 60) + 25;`

Correct Answer: D

Sample Free-Response Question: Methods and Control Structures

This question involves the use of *check digits*, which can be used to help detect if an error has occurred when a number is entered or transmitted electronically. An algorithm for computing a check digit, based on the digits of a number, is provided in part (a).

The `CheckDigit` class is shown at right. You will write two methods of the `CheckDigit` class.

(a) Complete the `getCheck` method, which computes the check digit for a number according to the following rules.

- Multiply the first digit by 7, the second digit (if one exists) by 6, the third digit (if one exists) by 5, and so on. The length of the method's `int` parameter is at most six; therefore, the last digit of a six-digit number will be multiplied by 2.
- Add the products calculated in the previous step.
- Extract the check digit, which is the rightmost digit of the sum calculated in the previous step.

```
public class CheckDigit
{
    /** Returns the check digit for num, as described in part (a).
     *  Precondition: The number of digits in num is between one and
     *  six, inclusive.
     *            num >= 0
     */
    public static int getCheck(int num)
    {
        /* to be implemented in part (a) */
    }

    /** Returns true if numWithCheckDigit is valid, or false
     *  otherwise, as described in part (b).
     *  Precondition: The number of digits in numWithCheckDigit
     *  is between two and seven, inclusive.
     *            numWithCheckDigit >= 0
     */
    public static boolean isValid(int numWithCheckDigit)
    {
        /* to be implemented in part (b) */
    }

    /** Returns the number of digits in num. */
    public static int getNumberOfDigits(int num)
    {
        /* implementation not shown */
    }

    /** Returns the nthdigit of num.
     * Precondition: n >= 1 and n <= the number of digits in num
     */
    public static int getDigit(int num, int n)
    {
        /* implementation not shown */
    }

    // There may be instance variables, constructors, and methods not shown.
}
```

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

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

00558-040 (Updated January 2021)