

**COURSE TITLE: AP Computer Science A**

Level of Difficulty	Estimated Homework	Prerequisites
Difficult	0-30 Minutes	<b>District:</b> C or higher in Algebra 2 <b>Department Suggestion:</b> C or higher in Algebra 2

**Course Description:**

This course allows all students to explore programming through problem solving. It also provides extra time to explore non-AP topics that students find interesting and motivational such as graphical interfaces and image development. Students learn to design and implement computer-based solutions to problems in a variety of application areas. In solving these problems, students develop, select, and implement appropriate algorithms and data structures. The object-oriented paradigm is used to problem solve using the Java programming language. Students read, interpret, and instantiate standard Java library classes from the AP Java subset. Reading, understanding, and appropriately modifying large programs like those provided in the College Board labs and instructor created labs consisting of several classes and interacting objects are also requirements of this course. While the Wiley Cay Horstmann text is the main resource for students, the course employs a spiraling approach to concept presentation and exploration using student inspired programming explorations. Many programming topics are discovered and/or requested by students early in the course in order to solve motivational problems. Algorithmic thinking and problem solving is enhanced through the use of the Javabat website created by Nick Parlante of Stanford University. Topics are revisited throughout the course in a more formal way to reinforce key concepts and deepen student understanding. Discussion of social and ethical implications of programming and computer use are motivated by articles related to recent events. Two examples of articles used to inspire past discussions are listed below.

The strategies used in this course are listed here and numbered for future reference. The Socratic method (1) is used regularly to allow students to share their insights, expand their understanding, and explore new related ideas in the context of their current work. Weekly reading assignments (2) from the course resources are used to provide students with initial insights that are refined through probing questioning and discussion. Kinesthetic Learning Exercises (3) are used throughout the course to allow students to visualize and act out important algorithms. Computer animations (4) are also used from various websites to reinforce student understanding. Early in the course, relatively short programming assignments are given on a weekly basis to allow students to apply their knowledge to code creation working individually (5) and in small groups (6). Codingbat is used (7) to allow students to improve their algorithmic problem solving skills. As the course progresses, students are required to work on group programming projects (8) that include design exercises and code sharing. At the end of the course, each student is required to create a relatively large program (9) that illustrates their ability to apply the concepts learned in class and through individual exploration to the creation of usable code.

**Text and Supplemental Materials:**

Course at a Glance:

<https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-a-glance.pdf?course=ap-computer-science-a>

Cay Horstmann. Computing Concepts with Java Essentials. New York: Wiley, 2007.

Programming Labs. The College Board 2020.

<https://apcentral.collegeboard.org/courses/ap-computer-science-a/classroom-resources/lab-resource-page>

Nick Parlante, Stanford University, Codingbat: A free site of live Java coding problems to build coding skill.

<https://codingbat.com/java>

Online review and practice exam problems.

<https://runestone.academy/runestone/books/published/apcsareview/index.html>

**Supplemental Information:**

UC subject area “c”