Teaching AP® Computer Science A

CodeHS Professional Development Course

Details

- Course delivery method: Online delivery
- Contact Hours: Approximately 40 contact hours
- Prerequisite(s): NA
- Recommended: Best for individuals planning to teach high school computer science within a year of completing the course

Description

Learn how to lead a successful AP® Computer Science A class with CodeHS. The Teaching AP® Computer Science A professional development course prepares teachers to teach AP® Computer Science A with a focus on the Java programming language. Teachers will practice the skills they need to teach tricky concepts, debug programs, answer questions, and lead a blended classroom. Additionally, they will explore best practices for engaging and accessible computer science instruction. They will learn strategies for facilitating collaboration with all levels of participants in a blended computer science classroom. By the end of this course, participants will be empowered with the skills, pedagogical knowledge, and confidence to deliver a successful middle or high school Java computer science class.

Course Goals

Given the level of dependence on computers for our social, economic, and daily needs, learning a fundamental level of computer science concepts has become a basic literacy for full civic participation. Becoming fluent in a computer programming language supports participants having access to a broader range of potential employment, increases their potential compensation for work, and adds a creative outlet to their lives. Participants will understand the importance of providing access and quality computer science instruction to diverse learners. They will be prepared to teach a successful and engaging computer science course, with a strong understanding of content and how to facilitate student learning.

Course Objectives

1. Participants will recognize issues in computer science education related to access and equity, and they will explain why it is essential that children are exposed to high-quality and inclusive computer science instruction in secondary school.
2. Participants will design lessons and activities that introduce foundational programming topics.

Note: The online course is self-paced, so times listed are estimates for each module. Students may complete each module in more or less time than is noted.

3. Participants will create simple programs and projects that utilize foundational programming skills, including functions, control structures, and top-down design.
4. Participants will identify common misunderstandings of novice coders and develop best practices for teaching these concepts in their classrooms.
5. Participants will demonstrate effective debugging strategies to guide their classes in identifying mistakes and correcting their work.
6. Participants will develop their own blended classroom vision and approaches to facilitate learning and collaboration in their environment.
7. Participants will generate procedures and discuss strategies to proactively manage challenges in a blended classroom.

Programming Environment

Teachers write and run programs in the browser using the CodeHS editor.

More Information

Browse the content of this course at https://codehs.com/course/apcsapd/overview

Course Breakdown (40 hours)

Module	Description
Module 1: Welcome	This introductory module provides an overview of the CodeHS professional development experience. Educators will explore key goals for the course, examine strategies for planning and pacing their teaching, and reflect on access and equity in computer science education. This module also sets the foundation for goal-setting and long-term success in teaching computer science. Assessed on: ● Short Answer
Module 2: How CodeHS Works	In this module, educators will gain familiarity with the CodeHS platform. You'll learn how to navigate the teacher and student views, explore course structure, and effectively use tools for assigning content, grading student work, and monitoring progress. Assessed on: ● Quizzes ● Exercises ● Grading and Respond (to student exercises)
Module 3: What is AP® CSA?	This module covers all of the tools and resources on CodeHS and through the College Board for supporting AP CSA teachers. Major resources include the FRQ Center, practice exams, the review course,

Note: The online course is self-paced, so times listed are estimates for each module. Students may complete each module in more or less time than is noted.

	Java practice problems, AP classroom, how to use College Board reader summaries and rubrics, and more. Assessed on: ● Quizzes ● Short essays ● Coding exercises ● Creating Resources
Module 4: Tools and Resources for Teaching AP® CSA	This module covers all of the tools and resources on CodeHS and through the College Board for supporting AP CSA teachers. Major resources include the FRQ Center, practice exams, the review course, Java practice problems, AP classroom, how to use College Board reader summaries and rubrics, and more. Assessed on: ● Quizzes ● Short essays ● Creating Resources ● Teaching Strategies
Module 5: Planning Your First Month of AP® CSA	In this module, teachers will take a deep dive into the first unit of AP CSA. They'll learn key programming concepts that they will introduce to their students as well as pedagogical strategies they can use in their classroom. Assessed on: ● Quizzes ● Short essays ● Coding exercises ● Creating Resources
Module 6: Spotlight Lessons in AP® CSA	This module focuses on key lessons throughout the remainder of the course. Teachers will get a deeper understanding of these pivotal lessons, learn about common student misconceptions, and discover ways to better support their students through these lessons and concepts. Assessed on: ● Quizzes ● Short essays ● Exercises ● Unit Test
Module 7: Debugging in Java	Finding and fixing errors in programs is a fundamental skill for new and experienced programmers to develop. This module introduces debugging, the art and science of fixing broken programs. Students

Note: The online course is self-paced, so times listed are estimates for each module. Students may complete each module in more or less time than is noted.

	learn techniques for finding common code errors and effective debugging strategies to share with students. Assessed on: ● Quizzes ● Short essays ● Providing feedback to student programs, demonstrating how to coach studentsthrough effective debugging strategies ● Debugging difficult students programs
Module 8: After the AP® Exam	This brief module explores how teachers can effectively use the planning and instructional time after the AP Exam.
Module 9: Collaboration in Computer Science	Discover how to foster collaboration in your classroom. This module covers structured group work, pair programming, peer code reviews, and strategies for creating a collaborative learning environment while addressing potential challenges that may arise. Assessed on: ● Quizzes ● Short essays ● Coding exercises ● Creating Resources ● Teaching Strategies
Module 10: Effective Use of AI	In this module, teachers will learn the essential skills needed to effectively use AI as an instructional aide. Teachers will learn about the various AI tools available in the marketplace, explore how they can use AI in the CodeHS platform and identify techniques to use AI outside of CodeHS for coding assistance, differentiated instruction, and more. Assessed on: ● Short essays ● Creating Resources ● Teaching Strategies
Module 11: Course Wrap-Up	The Course Wrap-Up module includes an end-of-course survey and a course completion badge.

Assessment and Learning Resources

- To pass the course, participants must complete 85% of the course assignments and activities.
- Participants will receive formative feedback and are afforded re-doing assignments, as needed.

Note: The online course is self-paced, so times listed are estimates for each module. Students may complete each module in more or less time than is noted.

- Instructors will communicate to students and grade work through the CodeHS platform on a daily basis M-F.
- The CodeHS Learning Management System includes all assigned readings, articles, videos, quizzes, coding examples, coding exercises, and other assignments in each module's content area.

Note: The online course is self-paced, so times listed are estimates for each module. Students may complete each module in more or less time than is noted.