

AP Computer Science Principles (AP CSP)

Course Overview: AP CSP is a brand new AP course that offers a gentle introduction to Computer Science: the study of how computers solve problems. This is a true introductory course – **you don't have to know anything about Computer Science in order to take this class.** In fact, the less you know, the better. The goal of this course is to give you a firm background in several aspects of computer science and show you how computing can be a tool to help you solve problems in whatever your other interests are.

This is **not a Math class**. This is **not a Programming-Only class**. This is **not a Sit-and-Listen class**.

This is a *make something* class. This is a *debate something* class. This is a *present something* class.

Unit 1: The Internet	→	This unit investigates the challenges and technological discoveries that were made as computer scientists tried to develop a system for computers to communicate reliably and efficiently, which later became the modern Internet. As the unit progresses, we develop our understanding of the agreements that make up the TCP model of internet communication, and end with a research project to investigate modern advances in communication & internet technology.
Unit 2: Analyzing Data	→	We begin by extending the idea of agreed-upon communication protocols to develop the idea of agreed-upon file types for computers to represent data at the binary level. Strategies for storing text files, image files, and sound files are examined, with an emphasis on compression and meta-data. We then shift gears to look at <i>lots</i> of files at once, investigating strategies for working with 'Big Data' in a table. Strategies for sorting and filtering data are examined (such as Pivot Tables), as well as how to use visualizations effectively to make the data tell a story.
Unit 3: Algorithmic Thinking	→	This unit introduces a central idea to computer science: the solution to a computer science problem is not a single <i>answer</i> , but a <i>process</i> (or <i>algorithm</i>) that can be used to generate this answer repeatedly. We develop this idea of algorithmic thinking by simulating several machines that take very basic commands and use these machines to solve problems, culminating in our own drawing programs that can create designs with a small set of commands. Basic programming is first introduced in this unit using a variation of Javascript and Turtle Graphics commands.
<p>These next two units are taught at the <i>same time</i>. We'll bounce back and forth between topics on the left and topics on the right</p>		
Unit 4A: Personal Data		Unit 4B: Python Programming
<p>In an age where computers can fit in our phones, on our wrists, in our cars, or even in our refrigerators: we examine the digital footprint each of us leaves and how this data is used by the world around us. We focus on three guiding questions: (1) How is our data collected?, (2) How is our data used / Why is it useful?, and (3) How can we avoid this <i>without</i> completely disconnecting? We ask these three questions first of corporations and then of the government. In the process, we explore what data we've already made available through the devices and services we use, then examine the role of privacy policies and how advertising is the main economic force of the modern internet. We also explore how our data and metadata is used for surveillance, leading to an understanding of modern encryption and cyber-security practices. The unit culminates with a discussion of 'hacking', known cyberattacks, and modern cyberwarfare. Much of this unit is guided by current events and our own experiences with the devices we use and the data we make available. The AP Explore Task is completed during this unit as well.</p>		<p>We begin our formal introduction to programming using the Python programming language. We progress through the language by learning basic input/output and mathematical commands, if-statements, and string manipulation. Pair Programming is emphasized to help develop strategies to tackle these problems and avoid mistakes in our programs.</p>
Unit 5: Programming to Solve Problems	→	This unit finalizes our understanding of Python programming by formalizing looping structures, lists/arrays, and functions which we had seen before in our drawing unit. Using programming as a tool, we investigate several problems in mathematics, linguistics, and other disciplines to see how programming can help us solve larger problems. This unit also includes several programming-only days in preparation for the AP Performance Tasks.

AP CSP Big Ideas

Creativity • Abstraction • Data • Algorithms • Programming • The Internet • Global Impact

This is An AP Course. Here's What That Means

Grades & College Credit: AP stands for 'Advanced Placement' – it means that you can earn *college credit* for taking this class and passing an exam at the end. Most colleges accept this class as a General Education requirement, which means that even if you decide to go to college for something completely different (like Music or Biology or Architecture or Creative Writing), this class still counts towards helping you finish faster. This class is also *weighted*, which means that it can really help your GPA. In most classes, a B is worth a 3.0 – but, in this class, a B is worth a 4.0, which is the same as an A in most other classes.

What to Expect: AP doesn't always mean *harder* or *faster* or *more work*, and that's especially true for this course. However, we do have high expectations in how you behave as a student and how responsible you are for your own learning. You are expected to do each of the following for the *entire* school year:

Manage your time so you can finish assignments and be prepared for class discussions	Read the board, check your email, and communicate with your peers so you are prepared for class	Contribute positively when working in groups or with a partner
Speak up when you don't understand something, even if that means coming to office hours	Be able to work independently without getting distracted	Prepare for tests and quizzes by reviewing and studying your notes and assignments
Keep your Promises	Be In Class	Complete Your Work

The AP Exam: In order to earn your college credit, you need to successfully complete three tasks. Plenty of time will be given to you in the 2nd semester to work on all three of these.

The Create Performance Task	The Explore Performance Task	Multiple-Choice Exam
Using any of the programming languages described in this class (or even languages not included in this class), develop a program on a topic that interests you or that solves a particular problem. You are encouraged to collaborate with a partner in developing and writing your program, but you <i>must write at least one significant portion of the program on your own</i> . Your program must implement at least one algorithmic process related to your topic or problem, and at least one abstraction that helps you manage the complexity of your program.	You will select and investigate a computing innovation that has had (or could have) a significant impact on society, the economy, or culture. In investigating this innovation, you should report on both the positive and negative effects of this innovation and any data-related concern, such as data storage, data privacy, or data security. This task is most similar to a <i>research report</i> on a computing topic of your own interest.	This is a 74 question Multiple Choice exam on the Big Ideas of this course: Creativity, Abstraction, Data, Algorithms, Programming, The Internet, and Global Impact.
This task is worth 24% of your AP Score. It is due April 28th	This task is worth 16% of your AP Score. It is due April 28th	The exam is worth 60% of your AP Score. The exam is on May 5th

**If you are taking this course, you are expected to take the AP Exam.
If you already know you're not planning on taking the Exam, we should talk now.**

Grades

Programs, Projects, & Research Papers: **35%** | Quizzes & Tests: **25%** | Everything Else: **30%**
Taking the AP Exam: **10%**

AP CSP Big Ideas

Creativity • Abstraction • Data • Algorithms • Programming • The Internet • Global Impact