



AP: DAILY VIDEOS

AP Computer Science Principles

AP Daily is a series of on-demand, short videos—created by expert AP teachers and faculty—that can be used for in-person, online, and blended/hybrid instruction. These videos cover every topic and skill outlined in the AP Course and Exam Description and are available in AP Classroom for students to watch anytime, anywhere.

Big Idea 1: Creative Development

Video Title	Video Focus	Instructor
1.1: Daily Video 1 – Collaboration	Why collaboration is important for the industry and why you should do it too!	Jaeson Han
1.1: Daily Video 2 – Collaboration	Explore ways to collaborate in AP CSP.	Jaeson Han
1.2: Daily Video 1 – Program Function and Purpose	In this video, we'll learn that all computing innovations are developed for a purpose and this purpose guides the development of the program.	Daniel Bonarigo
1.2: Daily Video 2 – Program Function and Purpose	In this video, we'll learn that programs take inputs in a variety of fashions to influence the program's behavior.	Daniel Bonarigo
1.2: Daily Video 3 – Program Function and Purpose	In this video, we'll learn that a computer program is a collection of statements that are executed according to the rules of the programming language.	Daniel Bonarigo
1.3: Daily Video 1 – Program Design and Development	In this video, we'll learn that computer programs are developed with a systematic approach to achieve a desired purpose.	Daniel Bonarigo
1.3: Daily Video 2 – Program Design and Development	In this video, we'll learn that programs are developed by groups of people and oftentimes new programs are written by expanding on the programs written by others.	Daniel Bonarigo
1.3: Daily Video 3 – Program Design and Development	In this video, we'll learn that programmers use comments and/or documentation to explain the purpose of their program code to themselves, collaborators, and others.	Daniel Bonarigo
1.4: Daily Video 1 – Identifying and Correcting Errors	In this video, we'll learn that errors are an inevitable part of programming, but understanding the different types of errors is essential to being a proficient programmer.	Daniel Bonarigo
1.4: Daily Video 2 – Identifying and Correcting Errors	In this video, we'll learn that errors are an inevitable part of programming, but understanding how to correct errors is essential to being a proficient programmer.	Daniel Bonarigo
1.4: Daily Video 3 – Identifying and Correcting Errors	In this video, we'll learn that programmers need to come up with testing schemes to ensure programs are working as intended.	Daniel Bonarigo

Big Idea 2: Data

Video Title	Video Focus	Instructor
2.1: Daily Video 1 – Binary Numbers	Data comes in all forms on computing devices, but at their simplest level, data is represented by bits. We will explore how all data can be represented using just two numbers.	Angelina Dominguez
2.1: Daily Video 2 – Binary Numbers	This video will address how using bits to represent data is not without consequences and will look at overflow, round-off, and other errors.	Angelina Dominguez
2.1: Daily Video 3 – Binary Numbers	In this video, we will demonstrate and practice how to convert between binary numbers and decimal numbers. Additionally, we will compare and order binary numbers.	Angelina Dominguez
2.2: Daily Video 1 – Data Compression	In this video, we will look at how compressing data can be done in a variety of different ways. We will compare and contrast some compression algorithms.	Angelina Dominguez
2.3: Daily Video 1 – Extracting Information from Data	In this video, we will examine some of the potential issues that come with analyzing data.	Sandy Czajka
2.3: Daily Video 2 – Extracting Information from Data	In this video, we will discuss how to extract knowledge from data and metadata, being aware of the difference between causation and correlation.	Sandy Czajka
2.4: Daily Video 1 – Using Programs with Data	In this video, we will explore how to process data and use a program to create common algorithms for analyzing data.	Sandy Czajka
2.4: Daily Video 2 – Using Programs with Data	In this video, we will look at the steps of processing information so programmers can gain useful insight and knowledge about the data.	Sandy Czajka

Big Idea 3: Algorithms and Programming

Video Title	Video Focus	Instructor
3.1: Daily Video 1 – Variables and Assignments	Variables are abstractions used to represent and store data. We can use variables to store data as numbers, Booleans, lists, and strings.	Jamila Mitchell
3.1: Daily Video 2 – Variables and Assignments	We can change the value stored in a variable with the assignment operator. The most recent value assigned will be the value stored, which will impact the output of a program.	Jamila Mitchell
3.2: Daily Video 1 – Data Abstraction	In this video, we will explain how lists and strings are examples of data abstraction.	Sandy Czajka
3.2: Daily Video 2 – Data Abstraction	In this video, we will use lists as an abstraction to manage the complexity of a program. We will also examine the lists section of the exam reference sheet.	Sandy Czajka
3.2: Daily Video 3 – Data Abstraction	In this video, we will look at examples of programs with lists and how this manages the complexity of a program.	Sandy Czajka
3.3: Daily Video 1 – Mathematical Expressions	We will examine how algorithms can be constructed using combinations of sequencing, selection, and iteration to provide a finite set of instructions to accomplish a specific task.	Jamila Mitchell
3.3: Daily Video 2 – Mathematical Expressions	We will use expressions to produce a value and determine how statements in program code are executed. Based on this, we will implement algorithms that achieve a desired result.	Jamila Mitchell
3.3: Daily Video 3 – Mathematical Expressions	Arithmetic operators can be used in expressions to produce a value. We will apply the order of operations used in mathematics to evaluate expressions using arithmetic operators.	Jamila Mitchell
3.4: Daily Video 1 – Strings	In this video, we will implement and interpret programs with pseudocode that use strings.	Sandy Czajka
3.5: Daily Video 1 – Boolean Expressions	Relational operators can be used to evaluate relationships between two variables, expressions, or values. These comparisons result in a Boolean value of either true or false.	Jamila Mitchell
3.5: Daily Video 2 – Boolean Expressions	We can use logical operators to evaluate multiple conditions to produce a single Boolean value. These conditions can be either Boolean expressions or single Boolean values.	Jamila Mitchell
3.5: Daily Video 3 – Boolean Expressions	We will evaluate relationships using relational operators and Boolean expressions to produce a single Boolean value of either true or false.	Jamila Mitchell
3.6: Daily Video 1 – Conditionals	We will create algorithms using selection to choose from possible outcomes. With selection, we can select which statements to execute based on the outcome of a condition.	Jamila Mitchell
3.6: Daily Video 2 – Conditionals	We will implement conditional statements, or if statements, to choose between different blocks of code based on the outcome of a Boolean expression.	Jamila Mitchell
3.6: Daily Video 3 – Conditionals	We will determine the outcome of program code that uses conditional statements. We will explore how if and if-else statements can be used to choose which statements to execute.	Jamila Mitchell
3.7: Daily Video 1 – Nested Conditionals	We will determine the outcome of program code that uses nested conditional statements. When conditionals are nested, we select statements based on the outcome of multiple conditions.	Bradley Bearden
3.7: Daily Video 2 – Nested Conditionals	We will create algorithms that use nested conditional statements. When nesting conditional statements, we must adjust the logic and flow to meet the needs of our algorithm.	Bradley Bearden

Video Title	Video Focus	Instructor
3.8: Daily Video 1 – Iteration	We will represent algorithms using iteration with a flow chart or natural language. We can iterate (or repeat) a part of our algorithm as many times as needed to accomplish our goal.	Bradley Bearden
3.8: Daily Video 2 – Iteration	We will determine the outcome of program code that uses iteration. We will determine how many times to iterate based on the pseudocode provided.	Bradley Bearden
3.8: Daily Video 3 – Iteration	We will create algorithms that use iteration using pseudocode. Small adjustments in the pseudocode can have a major impact on the outcome of the iteration.	Bradley Bearden
3.9: Daily Video 1 – Developing Algorithms	We will determine the outcome of competing algorithms. We will compare and contrast the competing algorithms to better understand their outcomes.	Bradley Bearden
3.9: Daily Video 2 – Developing Algorithms	We will represent algorithms using a flowchart or natural language. Whether algorithms contain selection or iteration, they must be sequenced properly.	Bradley Bearden
3.9: Daily Video 3 – Developing Algorithms	We will create algorithms by modifying existing correct algorithms. We will explore advantages and concerns that arise when modifying existing correct algorithms.	Bradley Bearden
3.10: Daily Video 1 – Lists	In this video, we will use the pseudocode for lists on the exam reference sheet to implement an algorithm.	Sandy Czajka
3.10: Daily Video 2 – Lists	In this video, we will use the pseudocode for lists on the exam reference sheet to determine lines of code necessary to complete the algorithm.	Sandy Czajka
3.10: Daily Video 3 – Lists	In this video, we will use the pseudocode for lists on the exam reference sheet to interpret code to determine the output of given code.	Sandy Czajka
3.11: Daily Video 1 – Binary Search	In this video, we will look at how a Binary Search Algorithm is structured and operates.	Catherine Medina-DeVilliers
3.12: Daily Video 1 – Calling Procedures	We will determine the outcome of procedure calls. These procedures may have parameters and/or return values.	Bradley Bearden
3.12: Daily Video 2 – Calling Procedures	We will write procedure calls. These procedures may have parameters and/or return values.	Bradley Bearden
3.13: Daily Video 1 – Developing Procedures	We will develop procedural abstraction by writing procedures that may have parameters and/or return values.	Bradley Bearden
3.13: Daily Video 2 – Developing Procedures	We will discover the value of using procedures to manage complexity and make the program more readable.	Bradley Bearden
3.14: Daily Video 1 – Libraries	In this video, we'll learn to use procedures that allow programmers to draw upon existing code, allowing them to write programs more quickly and with more confidence.	Julie Alano
3.15: Daily Video 1 – Random Values	In this video, we will look at how to evaluate expressions that use random number generators.	Catherine Medina-DeVilliers
3.15: Daily Video 2 – Random Values	In this video, we will look at how to draw on existing code to generate random numbers.	Catherine Medina-DeVilliers
3.16: Daily Video 1 – Simulations	In this video, we will look at how we use computers to reflect real-world phenomena through the use of simulations.	Catherine Medina-DeVilliers
3.16: Daily Video 2 – Simulations	In this video, we will compare computer simulations to real-world phenomena.	Catherine Medina-DeVilliers

Video Title	Video Focus	Instructor
3.17: Daily Video 1 – Algorithmic Efficiency	In this video, we will estimate an algorithm’s efficiency and evaluate the difference between algorithms that run in reasonable time and those that do not.	Julie Alano
3.17: Daily Video 2 – Algorithmic Efficiency	In this video, we will identify situations where a heuristic solution may be more appropriate.	Julie Alano
3.18: Daily Video 1 – Undecidable Problems	In this video, we will investigate the existence of undecidable problems in computer science.	Julie Alano

Big Idea 4: Computer Systems and Networks

Video Title	Video Focus	Instructor
4.1: Daily Video 1 – The Internet	We will look at how computers send and receive information using packets.	Catherine Medina-DeVilliers
4.1: Daily Video 2 – The Internet	We will look at how the Internet behaves and protocols.	Catherine Medina-DeVilliers
4.2: Daily Video 1 – Fault Tolerance	In this video, we will evaluate different networks to identify vulnerabilities and determine the benefits of fault tolerance.	Julie Alano
4.2: Daily Video 2 – Fault Tolerance	In this video, we will look at different network solutions that are fault tolerant and how that facilitates the transfer of data.	Julie Alano
4.3: Daily Video 1 – Parallel and Distributed Computing	In this video, we will look at how parallel and distributed computing leverage multiple computers to solve complex problems more quickly or process large data sets.	Catherine Medina-DeVilliers

Big Idea 5: Impact of Computing

Video Title	Video Focus	Instructor
5.1: Daily Video 1 – Beneficial and Harmful Effects	We will investigate how an effect of a computing innovation can be both beneficial and harmful.	Jaeson Han
5.1: Daily Video 2 – Beneficial and Harmful Effects	We will investigate how a computing innovation can have an impact beyond its intended purpose.	Jaeson Han
5.2: Daily Video 1 – Digital Divide	What is the digital divide?	Jaeson Han
5.3: Daily Video 1 – Computing Bias	In this video, we'll learn about bias in computing.	Jaeson Han
5.4: Daily Video 1 – Crowdsourcing	In this video, we will explore the advantages of a computer's reach.	Jaeson Han
5.5: Daily Video 1 – Legal and Ethical Concerns	In this video, we will explore how the use of computing can raise legal and ethical concerns.	Angelina Dominguez
5.6: Daily Video 1 – Safe Computing	In this video, we will look at the risks to privacy from collecting and storing personal data on a computer.	Angelina Dominguez
5.6: Daily Video 2 – Safe Computing	In this video, we will explore how computing resources can be misused as well as how they can be protected.	Angelina Dominguez
5.6: Daily Video 3 – Safe Computing	In this video, we will look at how unauthorized access to computing resources is gained.	Angelina Dominguez