UTeach Computer Science

AP Computer Science Principles
Syllabus and Planning Guide
(2023–2024)

Table of Contents

Curricular Requirements

Curricular Requirements		Pages
CR-1	The teacher and students have access to college-level computer science resources, in print or electronic format.	3
CR-2	The course provides opportunities to develop student understanding of the required content outlined in each of the Big Ideas described in the AP Course and Exam Description.	9–10
CR-3	The course provides opportunities to develop student understanding of the Big Ideas.	7–8
CR-4	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 1: Computational Solution Design.	5
CR-5	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 2: Algorithms and Program Development.	6
CR-6	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 3: Abstraction in Program Development.	6
CR-7	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 4: Code Analysis.	6
CR-8	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 5: Computing Innovations.	6
CR-9	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 6: Responsible Computing.	6
CR-10	The course provides a minimum of three opportunities for students to investigate different computing innovations.	35
CR-11	Students are provided at least twelve (12) hours of dedicated class time to complete the AP Create Performance Task.	39–40

Curriculum Description

Developers

UTeach CS Principles has been developed by The UTeach Institute ([UTeach Computer Science](#)) through a grant from the [National Science Foundation](#) (award #1543014).

Curriculum Overview

UTeach CS Principles is a year-long high school curriculum that fully addresses the five "Big Ideas" of computer science and six "Computational Thinking Practices," as specified by the College Board's AP Computer Science Principles curriculum framework.

The lessons and materials used throughout this curriculum incorporate Project-Based Learning (PBL), a pedagogical approach that actively engages students in the educational process, improves retention, and develops problem-solving, critical-thinking, and group communication skills. Through this collaborative, learner-centric approach, students are encouraged to explore the advantages and societal impact of computational technology while developing their own programming and computational thinking skills. *It is required for students have daily access to the internet.*

Textbook

[CR-1] UTeach CS Principles has an interactive online textbook available for students and teachers. The textbook, assessments, projects, and built-in programming environments are hosted on the Codio platform, which can be integrated with most learning management systems and is FERPA compliant.

Bibliography

Abelson, H., Ledeen, K., and Lewis, H. R. *Blown to Bits: Your life, liberty, and happiness after the digital explosion.* Upper Saddle River, N.J.: Addison-Wesley, 2008.

Programming Language Requirements

Throughout the curriculum, students will explore the coding process through the context of two different programming environments – Scratch and Python. Each of these platforms has been designed to provide beginning students with a simplified and novice-friendly interface with which to write their first dynamic and highly engaging programs. The tools for both environments are embedded alongside the student textbook within the Codio platform and do not require any additional software or licenses.

Scratch Programming Language (block-based)

Developed by the MIT Media Lab, Scratch offers students an introduction to coding using a visual programming interface. By dragging and dropping labeled programming components (a.k.a. "blocks") that snap together into syntactically correct compositions, students can quickly construct robust and fully functional programs with very little prior programming knowledge or skill. This block-based programming environment is ideally suited to first-time programmers as it abstracts away much of the low-level implementation details and allows students to clearly focus on the more generalized concepts that are so fundamental to the art of computational thinking. Students will create Scratch programs through an integrated development environment (IDE) embedded in Codio's platform, without needing to create separate Scratch accounts.

Python Programming Language (text-based)

Students transition to programming in a text-based language in the Python Bootcamp module in between units 3 and 4. Python is easy for teachers and students to learn, while also being well-suited for the Create Task and more widely established as an industry-standard language. Students will program in Python through the integrated development environment (IDE) within Codio's platform.

Course Framework

The course framework consists of two components: 1) Computational Thinking Practices, and 2) Course Content, which includes Big Ideas, Enduring Understandings, Learning Objectives, and Essential Knowledge Statements.

Computational Thinking Practices

Computational Thinking Practices: Skills [CR 4-9]					
1 Computational Solution Design: Design and evaluate computational solutions for a purpose.	**2** Algorithms and Program Development: Develop and implement algorithms.	**3** Abstraction in Program Development: Develop programs that incorporate abstractions.	**4** Code Analysis: Evaluate and test algorithms and programs.	**5** Computing Innovations: Investigate computing innovations.	**6** Responsible Computing: Contribute to an inclusive, safe, collaborative, and ethical computing culture.
1.A Investigate the situation, context or task.	**2.A** Represent algorithmic processes without using a programming language.	**3.A** Generalize data sources through variables.	**4.A** Explain how a code segment or program functions.	**5.A** Explain how computing systems work.	**6.A** Collaborate in the development of solutions.
1.B Determine and design an appropriate method or approach to achieve the purpose.	**2.B** Implement and apply an algorithm.	**3.B** Use abstraction to manage complexity in a program.	**4.B** Determine the result of code segments.	**5.B** Explain how knowledge can be generated from data.	**6.B** Use safe and secure methods when using computing devices.
1.C Explain how collaboration affects the development of a solution.		**3.C** Explain how abstraction manages complexity.	**4.C** Identify and correct errors in algorithms and programs, including error discovery through testing.	**5.C** Describe the impact of a computing innovation.	**6.C** Acknowledge the intellectual property of others.
1.D Evaluate solution options.				**5.D** Describe the impact of gathering data.	
				5.E Evaluate the use of computing based on legal and ethical factors.	

The following are examples in the curriculum of an instructional approach or activity that describes how students will engage with these skills:

CR-4: **1.B** Students will write a program to simulate the game of Rock, Paper, Scissors using two sprites for the players and a sprite for the referee. Students will design the solution using relational and logical operators to determine the winner of each round and the winner of the game.

CR-5: `2.A` For the Unit 1 Password Generator Project, students will work together in groups to create an algorithm that generates a unique and secure password for different websites. Students work together to use natural language to develop an algorithm that incorporates sequencing, selection, and iteration. Students also practice creating algorithms to move a robot around a grid using both the AP Exam Reference pseudocode robot procedures and blocks in the Scratch programming language.

CR-6: `3.B` In the Python Bootcamp module, students will create a program in Python that uses lists to store and analyze data. Students will write several of the common list processing algorithms such as determining a maximum or minimum value of the list, computing the sum of the values in the list, and displaying only the even numbers in the list.

CR-7: `4.A` In the Unit 4 Image Filter Project, students are to create a program that will apply filters to images. As part of their submission, students are to include an explanation of the code segment for each filter. Students are required to give a detailed explanation of their code that mathematically manipulates the RGB channels for each pixel to create the filter and transform the image.

CR-8: `5.C` Students will complete the Analyzing Impact of Computing in the College Board's AP Computer Science Principles Explore Curricular Requirements Teacher Resources in Unit 6.

CR-9: `6.A` For each of the unit projects, students work in either groups or pairs. On the first day of the unit when the project is launched, students complete a group contract in which they agree upon group roles and strategies for working together and collaborating over the unit project assignment.

Course Content - Big Ideas

The Big Ideas serve as the foundation of the course. They are overarching concepts or themes that are spiraled throughout all six of the curriculum units and connected to the topics and activities within each unit.

Big Ideas [CR 3]	
Big Idea 1: CREATIVE DEVELOPMENT (CRD) When developing computing innovations, developers can use a formal, iterative design process or experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool to use at any phase of development because considering multiple perspectives allows for improvement of innovations.	Each of the six unit projects encourage students to work independently and/or collaboratively in more open-ended, student-directed, hands-on projects and activities. The Unit 2 project allows students to use pair programming to design a program using the Scratch programming language. Students collaborate with a partner and use an iterative and incremental design process to create a program and user interface of their choosing. Program requirements are to use sequencing, selection, and iteration as well as user interaction and variables.
Big Idea 2: DATA (DAT) Data is central to computing innovations because it communicates initial conditions to programs and represents new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of this data. Computers store data digitally, which means that the data must be manipulated to be presented in a useful way to the user.	The Big Idea of Data is woven through all the units but Units 3 and 5 take a closer look at digital data. In Unit 3, students study binary and how all information is stored as bits. Students write a Binary Birthday Cake program in which they convert a decimal age to binary. In Unit 5 students take a deep dive into big data—how it is collected, extracted, and processed to gain new knowledge and insight. Students will take a publicly available data set and analyze the data to provide new information.
Big Idea 3: ALGORITHMS AND PROGRAMMING (AAP) Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction, by breaking problems down into interacting pieces, each with their own purpose, makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.	While students are expected to actively employ computational thinking techniques and practices throughout all their work, every unit has an activity in which students are to employ algorithmic thinking practices to demonstrate their programming skills. Example programming assignments from Unit 2 include creating a Math Mashup program that uses mathematical expressions, a Quiz Show that uses selection structures, and a Regular Polygon Generator program that includes iteration and modularity. In Unit 3, students create a program using the list data structure to store multiple pieces of related information.

Big Idea 4: COMPUTING SYSTEMS AND NETWORKS (CSN)	All the enduring understandings, learning objectives, and essential knowledge statements for the CSN Big
Computer systems and networks are used to transfer data. One of the largest and most used networks is the internet. Through a series of protocols, the internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer, but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.	Idea are covered thoroughly in Unit 6. Students will take a peek under the hood of the inner workings of the internet and computing systems and how data is transferred through networks. Students participate in several unplugged activities such as the Routing Activity, designed to simulate a packet traveling through a network, and the DNS Lookup game, which simulates how the Domain Name System turns a queried domain name into an IP address. Students will also research and report on a specific malware and how to protect oneself from personal security threats.
Big Idea 5: IMPACT OF COMPUTING (IOC)	Midway through each unit, students are asked to step
Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand how our programs will be used and be responsible for the consequences. As computer users, we need to understand how to protect ourselves and our privacy when using a computer.	back and consider the broader implications of the unit's main topic and its impact on society at large. Through in-class discussions, debates, and creative activities, students will extrapolate from the ideas and concepts presented in class to explore the implications of the use of and advances in computational technology. In Unit 6, students will study three different computing innovations and the impact they have had on society through the College Board's AP Computer Science Principles Explore Curricular Requirement Teacher Resources activities.

Unit Guides

The year-long curriculum consists of six instructional units, a Python Bootcamp unit, and a unit for the development of the Create Performance Task that have been carefully structured to gently guide novice students through the study of computational technology by first establishing a context for the curriculum material, then teaching the core skills for creating and using computational tools, followed by demonstrating real-world applications of digital technology, and finally allowing the students to exhibit the skills they have developed.

Curriculum Units [CR 2]		
Unit 1: Computational Thinking Discover computational thinking, logical reasoning, and describing processes through algorithms and pseudocode.	**Big Ideas:** Creative Development Data Algorithms and Programming Impact of Computing	**Computational Thinking Practices (CTP):** 1 2 3 5 **Enduring Understandings (EU):** CRD-1, DAT-2, AAP-2, AAP-4 IOC-1, IOC-2
Unit 2: Programming Use *Scratch* to explore sequencing, selection, and iteration as part of the goal to create programs that serve useful functions.	**Big Ideas:** Creative Development Data Algorithms and Programming Impact of Computing	**Computational Thinking Practices (CTP):** 1 2 3 4 5 6 **Enduring Understandings (EU):** CRD-1, CRD-2, DAT-1, AAP-1, AAP-2, AAP-3, IOC-1
Unit 3: Data Representation Explore the different means of representing information digitally.	**Big Ideas:** Creative Development Data Algorithms and Programming Impact of Computing	**Computational Thinking Practices (CTP):** 1 2 3 4 5 **Enduring Understandings (EU):** CRD-2, DAT-1, AAP-1, AAP-2, IOC-1
Python Bootcamp Module Learn the text-based programming language, *Python*.	**Big Ideas:** Creative Development Algorithms and Programming	**Computational Thinking Practices (CTP):** 1 2 3 4 **Enduring Understandings (EU):** CRD-2, AAP-1, AAP-2, AAP-3

	Big Ideas:	Computational Thinking Practices (CTP):

Unit 4: Digital Media Processing
 	
Use *Python* to programmatically manipulate digital images and audio.
	Big Ideas:
	   Creative Development
	   Data
	   Algorithms and Programming
	   Impact of Computing
		Computational Thinking Practices (CTP):
		1 2 3 4 5
		
		Enduring Understandings (EU):
		   CRD-1, DAT-1, AAP-1, AAP-2, IOC-1

Unit 5: Big Data
 	
Discover new knowledge using large data sets.
	Big Ideas:
	   Data
	   Algorithms and Programming
	   Impact of Computing
		Computational Thinking Practices (CTP):
		1 2 5
		
		Enduring Understandings (EU):
		   DAT-2, AAP-3, IOC-1, IOC-2

Unit 6: Innovative Technologies
 	
Explore the current state of technology and its role in our everyday lives.
	Big Ideas:
	   Creative Development
	   Data
	   Algorithms and Programming
	   Computing Systems and Networks
	   Impact of Computing
		Computational Thinking Practices (CTP):
		1 2 3 5
		
		Enduring Understandings (EU):
		   CRD-1, CRD-2, DAT-2, AAP-1, CSN-1, CSN-2, IOC-1, IOC-2

Create Task Demonstrate their learning by completing the Create Performance Task for submission to the College Board.	**Summative Assessment**	

Unit 1: Computational Thinking

To successfully master the art of creating computational artifacts, it is important that students develop a clear understanding of the complex processes and structures that make up an algorithmic solution to a given problem. In addition, it is critical that they be able to formally express those solutions clearly and unambiguously, such as what can be achieved using pseudocode or a well-specified programming language. This unit focuses on introducing students to these concepts and helping them to develop the skills that they will rely on throughout the remainder of the course.

First, students will explore several techniques for analyzing common problems and visualizing their solutions. They will use these techniques to investigate several real-world applications, such as searching, sorting, and encryption. Next, students will examine how programmers utilize various levels of abstraction in the languages that they use to write programs and communicate their intentions in a form that can be executed by a computer. Finally, students will turn their attention to the question of whether various problems are solvable and investigate the factors that affect the efficiency of a solution to a given problem.

The Big Ideas of Creative Development, Data, Algorithms and Programming, and Impact of Computing are covered in Unit 1. These are the Enduring Understandings covered in this unit:
- CRD-1: Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- DAT-2: Programs can be used to process data, which allows users to discover information and create new knowledge.
- AAP-2: The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- AAP-4: There exists problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.
- IOC-2: The use of computing innovations may involve risks to personal safety and identity.

These are the Computational Thinking Skills addressed:

- 1.A Investigate the situation, context, or task.
- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 1.C Explain how collaboration affects the development of a solution.
- 1.D Evaluate solution options.
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.
- 4.B Determine the result of code segments.
- 5.D Describe the impact of a gathering data.
- 5.E Evaluate the use of computing based on legal and ethical factors.
- 6.A Collaborate in the development of solutions.

Unit 1 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.1 Collaboration
- 1.3 Program Design and Development
- 2.3 Extracting Information from Data
- 3.3 Mathematical Expressions
- 3.6 Conditionals
- 3.8 Iteration
- 3.9 Developing Algorithms
- 3.11 Binary Search
- 3.17 Algorithm Efficiency
- 3.18 Undecidable Problem
- 5.3 Computing Bias
- 5.5 Legal and Ethical Concerns
- 5.6 Safe Computing

Unit 1 Schedule (18 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
1.1	Password Generator Project (1 of 5)	CRD-1.C, IOC-2.B	CRD-1.C.1, IOC-2.B.2
1.2	Algorithmic Thinking	AAP-2.A, AAP-2.B, AAP-2.G, AAP-2.H, AAP-2.J, AAP-2.K, AAP-2.M	AAP-2.A.1, AAP-2.A.2, AAP-2.A.3, AAP-2.A.4, AAP-2.B.1, AAP-2.G.1, AAP-2.H.1, AAP-2.J.1, AAP-2.K.1, AAP-2.M.1
1.3	Algorithmic Components: Sequencing, Selection, and Iteration	AAP-2.A, AAP-2.B, AAP-2.G, AAP-2.H, AAP-2.J, AAP-2.K, AAP-2.M	AAP-2.A.1, AAP-2.A.2, AAP-2.A.3, AAP-2.A.4, AAP-2.B.1, AAP-2.G.1, AAP-2.H.1, AAP-2.J.1, AAP-2.K.1, AAP-2.M.1
1.4	Encryption	AAP-2.A, AAP-2.G, AAP-2.H, AAP-2.J, AAP-2.K, AAP-2.M, IOC-2.B	AAP-2.A.1, AAP-2.A.2, AAP-2.A.4, AAP-2.G.1, AAP-2.H.1, AAP-2.J.1, AAP-2.K.1, AAP-2.M.1, IOC-2.B.5
1.5	Vigenère Cipher	AAP-2.A, AAP-2.G, AAP-2.H, AAP-2.J, AAP-2.K, AAP-2.M, IOC-2.B	AAP-2.A.1, AAP-2.A.2, AAP-2.A.4, AAP-2.G.1, AAP-2.H.1, AAP-2.J.1, AAP-2.K.1, AAP-2.M.1, IOC-2.B.5
1.1	Password Generator Project (2 of 5)	CRD-1.C, IOC-2.B	CRD-1.C.1, IOC-2.B.2
1.6	Programming Languages	CRD-2.E, CRD-2.F, AAP-2.A, AAP-2.M	CRD-2.E.1, CRD-2.E.2, CRD-2.E.3, CRD-2.E.4, CRD-2.F.2, CRD-2.F.3, CRD-2.F.4, CRD-2.F.5, CRD-2.F.6, AAP-2.A.1, AAP-2.A.2, AAP-2.A.3, AAP-2.A.4, AAP-2.M.1

1.1	Password Generator Project (3 of 5)	CRD-1.C, IOC-2.B	CRD-1.C.1, IOC-2.B.2
1.7	Coding Skills: Pseudocode	AAP-2.A, AAP-2.M	AAP-2.A.2, AAP-2.M.2
1.8	Decidability and Performance	AAP-2.L, AAP-2.O, AAP-2.P, AAP-4.A, AAP-4.B, CSN-1.B	AAP-2.L.1, AAP-2.L.2, AAP-2.L.5, AAP-2.O.5, AAP-2.P.1, AAP-2.P.2, AAP-2.P.3, AAP-4.A.1, AAP-4.A.2, AAP-4.A.3, AAP-4.A.4, AAP-4.A.5, AAP-4.A.6, AAP-4.A.7, AAP-4.B.1, AAP-4.B.2, AAP-4.B.3, CSN-1.B.6
1.9	Heuristics	AAP-4.A	AAP-4.A.8, AAP-4.A.9
1.10	Big Picture: Algorithmic Bias	CRD-1.A, DAT-2.C, IOC-1.D, IOC-1.F	CRD-1.A.4, DAT-2.C.5, IOC-1.D.1, IOC-1.D.2, IOC-1.D.3, IOC-1.F.11
1.1	Password Generator Project (4 of 5)	CRD-1.C, IOC-2.B	CRD-1.C.1, IOC-2.B.2
1.11	Unit 1 Review	None	None
1.12	Unit 1 Exam	None	None
1.1	Project Presentations (5 of 5)	None	None

Unit 2: Programming

When used correctly, computational technologies can prove be extremely powerful and effective tools for solving a wide range of problems. But to fully harness that power, an individual needs to be proficient in instructing those tools to perform highly precise operations in well-structured and logical sequences. This unit seeks to ease students into this new, structured, and more formalized way of thinking about problem solving and programming using Scratch, a block-based, visual programming language.

Once introduced to the Scratch IDE within Codio, students will experiment with several basic programming concepts and constructs, such as variables, user input, and selection statements. In the process, students will not only learn how to implement intended functionality by constructing well-designed blocks of executable code, but they will also explore techniques for debugging their code and verifying its correctness.

The Big Ideas of Creative Development, Data, Algorithms and Programming, and Impact of Computing are all covered in Unit 2. These are the Enduring Understandings covered in this unit:

- CRD-1: Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- DAT-1: The way a computer represents data internally is different from the way data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.
- AAP-1: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- AAP-2: The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- AAP-3: Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

These are the Computational Thinking Skills addressed:

- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 1.C Explain how collaboration affects the development of a solution.
- 1.D Evaluate solution options.
- 2.B Implement and apply an algorithm.
- 3.A Generalize data sources through variables.
- 3.B Use abstraction to manage complexity.
- 3.C Explain how abstraction manages complexity.
- 4.A Explain how a code segment or program functions.
- 4.B Determine the result of code segments.
- 5.E Evaluate the use of computing based on legal and ethical factors.
- 6.A Collaborate in the development of solutions.
- 6.C Acknowledge the intellectual property of others.

Unit 2 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.1 Collaboration
- 1.2 Program Function and Purpose
- 1.3 Program Design and Development
- 2.1 Binary Numbers
- 3.1 Variables and Assignment
- 3.3 Mathematical Expressions
- 3.5 Boolean Expressions
- 3.6 Conditionals
- 3.7 Nested Conditionals
- 3.8 Iteration
- 3.12 Calling Procedures
- 3.15 Random Values
- 5.5 Legal and Ethical Concerns

Unit 2 Schedule (23 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
2.1	Scratch Programming Project (1 of 6)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.A, AAP-2.B, AAP-2.H, AAP-2.K, IOC-1.F	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3. CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4, CRD-2.F.7, CRD-2.G.1, CRD-2.G.2, CRD-2.G.3, CRD-2.G.4, CRD-2.G.5, CRD-2.H.1, CRD-2.H.2, IOC-1.F.4, IOC-1.F.6
2.2	Welcome to Scratch	AAP-2.B	AAP-2.B.1, AAP-2.B.2, AAP-2.B.6
2.3	Robot Maze Algorithms	CRD-2.C, AAP-2.B	CRD-2.C.2, CRD-2.C.3, CRD-2.C.5, AAP-2.B.1, AAP-2.B.2, AAP-2.B.6
2.4	Let's Get Animated	CRD-2.C, CRD-2.G, AAP-2.B	CRD-2.C.2, CRD-2.C.3, CRD-2.C.5, CRD-2.G.1, CRD-2.G.2, CRD-2.G.3, CRD-2.G.4, AAP-2.B.1, AAP-2.B.2, AAP-2.B.6
2.5	Coding Skills: Animated Movie	AAP-2.B	
2.1	Scratch Programming Project (2 of 6)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.A, AAP-2.B, AAP-2.H, AAP-2.K, IOC-1.F	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3. CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4, CRD-2.F.7, CRD-2.G.1, CRD-2.G.2, CRD-2.G.3, CRD-2.G.4, CRD-2.G.5, CRD-2.H.1, CRD-2.H.2, IOC-1.F.4, IOC-1.F.6

2.6	User Input & Variables	CRD-2.C, DAT-1.A, AAP-1.A	CRD-2.C.1, CRD-2.C.4, CRD-2.C.6, DAT-1.A.1, AAP-1.A.1, AAP-1.A.2
2.7	Mathematical Expressions	AAP-2.C, AAP-3.E	AAP-2.C.1, AAP-2.C.3, AAP-3.E.1, AAP-3.E.2
2.8	Coding Skills: Math Mashup	AAP-1.A, AAP-2.C, AAP-3.E	
2.9	Selection Statements	AAP-2.E, AAP-2.F, AAP-2.H, AAP-2.I AAP-2.L	AAP-2.E.1, AAP-2.F.1, AAP-2.F.2, AAP-2.F.3, AAP-2.F.4, AAP-2.F.5, AAP-2.H.1, AAP-2.I.1, AAP-2.L.3, AAP-2.L.4
2.10	Game of Tag	AAP-2.E, AAP-2.F, AAP-2.H, AAP-2.I, AAP-2.L	
2.11	Coding Skills: Quiz Show	AAP-2.H	
2.1	Scratch Programming Project (3 of 6)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.A, AAP-2.B, AAP-2.H, AAP-2.K, IOC-1.F AAP-2.B, AAP-2.H	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3. CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4, CRD-2.F.7, CRD-2.G.1, CRD-2.G.2, CRD-2.G.3, CRD-2.G.4, CRD-2.G.5, CRD-2.H.1, CRD-2.H.2, IOC-1.F.4, IOC-1.F.6
2.12	Iteration	CRD-2.D, AAP-2.K	CRD-2.D.1, AAP-2.K.1, AAP-2.K.4, AAP-2.K.5
2.13	Coding Skills: Iteration	CRD-2.D, AAP-2.K	
2.14	Procedures	DAT-1.A, AAP-3.A	DAT-1.A.1, DAT-1.A.5, AAP-3.A.1, AAP-3.A.3, AAP-3.A.4
2.11	Coding Skills: Quiz Show with Procedures	DAT-1.A, AAP-3.A	
2.15	Coding Skills: Rock, Paper, Scissors	AAP-2.F, AAP-2.H, AAP-2.I	

2.1	Scratch Programming Project (4 & 5 of 6)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.A, AAP-2.B, AAP-2.H, AAP-2.K, IOC-1.F	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3. CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4, CRD-2.F.7, CRD-2.G.1, CRD-2.G.2, CRD-2.G.3, CRD-2.G.4, CRD-2.G.5, CRD-2.H.1, CRD-2.H.2, IOC-1.F.4, IOC-1.F.6
2.16	Unit 2 Review	None	None
2.17	Unit 2 Exam	None	None
2.1	Project Presentations (6 of 6)	None	None

Unit 3: Data Representation

To make the most effective use of computational tools and data-driven applications, students need to have a clear awareness and sense of comfort with the diverse kinds of information that may be available for use by these programs and the various ways that information may be digitally represented, stored, and manipulated within the computer. This unit focuses on providing students with an overview of the various levels of abstraction that are used in the digital representation of discrete data and information.

Students will initially focus on the lowest levels of digital representation and storage by examining different base representations of numbers (including decimal and binary) and their application to ASCII and Unicode character encoding. Students will also explore the distinctions between analog and digital forms of representation. Finally, students will examine the characteristics of lists and the types of common use-cases for these linear, ordered collections, including traversing, searching, and sorting.

The Big Ideas of Creative Development, Data, Algorithms and Programming, and Impact of Computing are all covered in Unit 3. These are the Enduring Understandings covered in this unit:
- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- DAT-1: The way a computer represents data internally is different from the way data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.
- AAP-1: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- AAP-2: The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

These are the Computational Thinking Skills addressed:
- 1.A Investigate the situation, context, or task.
- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 1.D Evaluate solution options.
- 2.B Implement and apply an algorithm.
- 3.B Use abstraction to manage complexity.
- 3.C Explain how abstraction manages complexity.
- 4.B Determine the result of code segments.
- 5.E Evaluate the use of computing based on legal and ethical factors.

Unit 3 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.4 Identifying and Correcting Errors
- 2.1 Binary Numbers
- 3.2 Data Abstraction
- 3.3 Mathematical Expressions
- 3.10 Lists
- 3.11 Binary Search
- 5.5 Legal and Ethical Concerns

Unit 3 Schedule (20 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
3.1	Unintend'o Controller Project (1 of 5)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.C, AAP-1.D, AAP-2.B, AAP-2.H, AAP-2.K, AAP-2.N, AAP-2.O, DAT-1.A	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3, CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4
3.2	Binary Encoding	DAT-1.A, DAT-1.C, AAP-2.P	DAT-1.A.2, DAT-1.A.3, DAT-1.A.4, DAT-1.A.6, DAT-1.A.7, DAT-1.C.1, DAT-1.C.2, DAT-1.C.3, DAT-1.C.4, DAT-1.C.5, AAP-2.P.1
3.3	Base Conversions	DAT-1.C	DAT-1.C.1, DAT-1.C.2, DAT-1.C.3, DAT-1.C.4, DAT-1.C.5
3.4	Coding Skills: Binary Birthday Cake	CRD-2.J, AAP-2.C	CRD-2.J.1, CRD-2.J.2, CRD-2.J.3, AAP-2.C.1, AAP-2.C.2, AAP-2.C.3, AAP-2.C.4
3.5	Common Encoding Standards	DAT-1.A, DAT-1.B	DAT-1.A.2, DAT-1.A.5, DAT-1.A.6, DAT-1.A.7, DAT-1.B.1, DAT-1.B.2, DAT-1.B.3
3.1	Unintend'o Controller Project (2 of 5)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.C, AAP-1.D, AAP-2.B, AAP-2.H, AAP-2.K, AAP-2.N, AAP-2.O, DAT-1.A	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3, CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4
3.6	Digitization	DAT-1.A	DAT-1.A.9
3.7	Discrete vs. Continuous	DAT-1.A	DAT-1.A.9
3.8	Big Picture: Legality of Reselling Digital Music	IOC-1.F	IOC-1.F.11

3.1	Unintend'o Controller Project (3 of 5)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.C, AAP-1.D, AAP-2.B, AAP-2.H, AAP-2.K, AAP-2.N, AAP-2.O, DAT-1.A	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3, CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4
3.9	Introduction to Lists	DAT-1.A, AAP-1.C, AAP-1.D	DAT-1.A.1, AAP-1.C.1, AAP-1.C.2, AAP-1.C.3, AAP-1.D.2, AAP-1.D.3, AAP-1.D.4, AAP-1.D.5, AAP-1.D.6, AAP-1.D.8
3.10	Processing a List	DAT-1.A, AAP-1.C, AAP-1.D, AAP-2.N, AAP-2.O	DAT-1.A.1, AAP-1.C.1, AAP-1.C.2, AAP-1.C.3, AAP-1.D.2, AAP-1.D.3, AAP-1.D.4, AAP-1.D.5, AAP-1.D.6, AAP-1.D.8, AAP-2.N.2, AAP-2.O.1, AAP-2.O.2
3.11	Coding Skills: Lists in Action	DAT-1.A, AAP-1.C, AAP-1.D, AAP-2.N, AAP-2.O	
3.12	Sorting Lists	DAT-1.A, AAP-1.C, AAP-1.D, AAP-2.N, AAP-2.O	DAT-1.A.1, AAP-1.C.1, AAP-1.C.2, AAP-1.C.3, AAP-1.D.2, AAP-1.D.3, AAP-1.D.4, AAP-1.D.5, AAP-1.D.6, AAP-1.D.8, AAP-2.N.2, AAP-2.O.1, AAP-2.O.2
3.1	Unintend'o Controller Project (4 of 5)	CRD-1.B, CRD-1.C, CRD-2.B, CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H, AAP-1.C, AAP-1.D, AAP-2.B, AAP-2.H, AAP-2.K, AAP-2.N, AAP-2.O, DAT-1.A	CRD-1.B.2, CRD-1.C.1, CRD-2.B.1, CRD-2.B.2, CRD-2.B.3, CRD-2.B.4, CRD-2.B.5, CRD-2.E.3, CRD-2.E.4
3.13	Unit 3 Review	None	None
3.14	Unit 3 Exam	None	None
3.1	Project Presentations (5 of 5)	None	None

Python Bootcamp

Building upon earlier visual programming experiences with Scratch, this module guides students through the transition to programming in a high-level, procedural language through a brief introduction to Python. Students will become familiar with a text-based environment that more closely reflects the actual programming tools used in industry and will be better equipped for continuing studies in computer science beyond the scope of this course.

The Big Ideas of Creative Development and Algorithms and Programming are covered in the Python Bootcamp unit. These are the Enduring Understandings covered in this unit:

- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- AAP-1: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- AAP-2: The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- AAP-3: Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

These are the Computational Thinking Skills addressed:

- 1.B Determine and design an appropriate method or approach to achieve the purpose.
- 2.A Represent algorithmic processes without using a programming language.
- 2.B Implement and apply an algorithm.
- 3.A Generalize data sources through variables.
- 3.B Use abstraction to manage complexity.
- 3.C Explain how abstraction manages complexity.
- 4.B Determine the result of code segments.
- 4.C Identify and correct errors in algorithms and programs, including error discovery through testing.

Python Bootcamp Topics

Over the course of this module, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.3 Program Design and Development
- 1.4 Identifying and Correcting Errors
- 3.1 Variables and Assignment
- 3.2 Data Abstraction
- 3.3 Mathematical Expressions
- 3.4 Strings
- 3.5 Boolean Expressions
- 3.6 Conditionals
- 3.8 Iteration
- 3.9 Developing Algorithms
- 3.10 Lists
- 3.12 Calling Procedures
- 3.13 Developing Procedures
- 3.14 Libraries

Python Bootcamp Schedule (12 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
PB.1	Introduction to Python in Codio		
PB.2	Scratch vs. Python		
PB.3	Python Basics	CRD-2.I	CRD-2.I.1, CRD-2.I.2, CRD-2.I.3, CRD-2.I.4, CRD-2.I.5
PB.4	User Input & Variables	AAP-1.A, AAP-1.B, AAP-1.D, AAP-2.D, AAP-2.E, AAP-3.A	AAP-1.A.3, AAP-1.A.4, AAP-1.B.1, AAP-1.B.2, AAP-1.B.3, AAP-1.D.1, AAP-2.D.1, AAP-2.D.2, AAP-2.E.1, AAP-3.A.6, AAP-3.A.9
PB.5	Selection Structures	AAP-2.E, AAP-2.H	AAP-2.E.2, AAP-2.H.2
PB.6	Libraries	AAP-3.D	AAP-3.D.1, AAP-3.D.2, AAP-3.D.3, AAP-3.D.4, AAP-3.D.5
PB.7	Iteration Structures	AAP-2.K	AAP-2.K.2, AAP-2.K.3
PB.8	Strings	AAP-1.C, AAP-2.D	AAP-1.C.4, AAP-2.D.1, AAP-2.D.2
PB.9	Lists	AAP-1.C, AAP-1.D, AAP-2.N	AAP-1.C.1, AAP-1.D.2, AAP-1.D.3, AAP-1.D.4, AAP-1.D.5, AAP-1.D.6, AAP-1.D.7, AAP-1.D.8, AAP-2.N.1, AAP-2.N.2
PB.10	Processing Lists	AAP-1.C, AAP-1.D, AAP-2.M, AAP-2.N, AAP-2.O	AAP-1.C.1, AAP-1.D.2, AAP-1.D.3, AAP-1.D.4, AAP-1.D.5, AAP-1.D.6, AAP-1.D.7, AAP-1.D.8, AAP-2.M.2, AAP-2.N.1,

			AAP-2.N.2, AAP-2.O.1, AAP-2.O.2, AAP-2.O.3, AAP-2.O.4
PB.11	Functions	AAP-2.B, AAP-3.A, AAP-3.B, AAP-3.C	AAP-2.B.7, AAP-3.A.1, AAP-3.A.2, AAP-3.A.3, AAP-3.A.4, AAP-3.A.5, AAP-3.A.7, AAP-3.A.8, AAP-3.B.1, AAP-3.B.2, AAP-3.B.3, AAP-3.B.4, AAP-3.B.5, AAP-3.B.6, AAP-3.B.7, AAP-3.C.1, AAP-3.C.2
PB.12	Parameters	AAP-2.B, AAP-3.A, AAP-3.B, AAP-3.C	AAP-2.B.7, AAP-3.A.1, AAP-3.A.2, AAP-3.A.3, AAP-3.A.4, AAP-3.A.5, AAP-3.A.7, AAP-3.A.8, AAP-3.B.1, AAP-3.B.2, AAP-3.B.3, AAP-3.B.4, AAP-3.B.5, AAP-3.B.6, AAP-3.B.7, AAP-3.C.1, AAP-3.C.2
PB.13	Coding Skills: Mastermind Game	CRD-2.F, AAP-2.B, AAP-2.C	

Unit 4: Digital Media Processing

Students will explore how digital media such as images and audio files are simply a series of bits that can be manipulated mathematically. Students will explore the characteristics of the RGB color model and its use in encoding digital images. For the unit project, they will apply these concepts toward the implementation of a series of algorithmic filters for digitally modifying images to achieve various visual effects. Finally, students will also investigate the methods of representing and modifying digital audio, including Auto-Tune and data compression.

The Big Ideas of Creative Development, Data, Algorithms and Programming, and Impact of Computing are all covered in Unit 4. These are the Enduring Understandings covered in this unit:

- CRD-1: Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- DAT-1: The way a computer represents data internally is different from the way data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.
- AAP-1: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- AAP-2: The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

These are the Computational Thinking Skills addressed:
- 3.C Explain how abstraction manages complexity.
- 5.E Evaluate the use of computing based on legal and ethical factors.

Unit 4 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.1 Collaboration
- 2.1 Binary Numbers
- 2.2 Data Compression
- 3.1 Variables and Assignment
- 3.2 Data Abstraction
- 3.6 Conditionals
- 3.8 Iteration
- 3.10 Lists
- 5.5 Legal and Ethical Concerns

Unit 4 Schedule (14 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
4.1	Image Filter Project (1 of 5)	CRD-1.C, AAP-1.C, AAP-1.D, AAP-2.H, AAP-2.K, AAP-2.O	
4.2	RGB Color	DAT-1.A	DAT-1.A.6, DAT-1.A.7
4.3	Raster Image Manipulation	DAT-1.A, AAP-1.C	DAT-1.A.6, DAT-1.A.7 AAP-1.C.1
4.1	Image Filter Project (2 of 5)	CRD-1.C, AAP-1.C, AAP-1.D, AAP-2.H, AAP-2.K, AAP-2.O	
4.4	Digitizing Audio	DAT-1.A	DAT-1.A.8, DAT-1.A.10
4.5	Audio Manipulation	DAT-1.A	DAT-1.A.8, DAT-1.A.10
4.1	Image Filter Project (3 of 5)	CRD-1.C, AAP-1.C, AAP-1.D, AAP-2.H, AAP-2.K, AAP-2.O	
4.6	Data Compression	DAT-1.D	DAT-1.D.1, DAT-1.D.2, DAT-1.D.3, DAT-1.D.4, DAT-1.D.5, DAT-1.D.6, DAT-1.D.7, DAT-1.D.8
4.7	Big Picture: Ethics of Digital Manipulation	IOC-1.F	IOC-1.F.1, IOC-1.F.2, IOC-1.F.3, IOC-1.F.5, IOC-1.F.7, IOC-1.F.8
4.8	Creative Commons	IOC-1.F	IOC-1.F.1, IOC-1.F.2, IOC-1.F.3, IOC-1.F.5, IOC-1.F.7, IOC-1.F.8
4.1	Image Filter Project (4 of 5)	CRD-1.C, AAP-1.C, AAP-1.D, AAP-2.H, AAP-2.K, AAP-2.O	
4.9	Unit 4 Review	None	None

4.10	Unit 4 Exam	None	None
4.1	Project Presentations (5 of 5)	None	None

Unit 5: Big Data

One of the most powerful applications of computational thinking relates to the creation and analysis of large data sets. In this unit, students will explore the complete set of processes and techniques that are involved in collecting large volumes of raw data and extracting new and useful information. Students will look at a variety of ways that data scientists use data-mining techniques to construct and visualize new knowledge. And finally, using these techniques themselves, students will perform their own analysis on a sample data set to discover new insights, which they will share with the class through a formal, TED-style presentation.

The Big Ideas of Data and Impact of Computing are all covered in Unit 5. These are the Enduring Understandings covered in this unit:

- DAT-2: Programs can be used to process data, which allows users to discover information and create new knowledge.
- AAP-3: Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.
- IOC-2: The use of computing innovations may involve risks to personal safety and identity.

These are the Computational Thinking Skills addressed:

- 1.A Investigate the situation, context, or task.
- 1.C Explain how collaboration affects the development of a solution.
- 1.D Evaluate solution options.
- 2.B Implement and apply an algorithm.
- 5.B Explain how knowledge can be generated from data.
- 5.D Describe the impact of a gathering data.
- 5.E Evaluate the use of computing based on legal and ethical factors.

Unit 5 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 2.3 Extracting Information from Data
- 2.4 Using Programs with Data
- 5.4 Crowdsourcing
- 5.5 Legal and Ethical Concerns
- 5.6 Safe Computing

Unit 5 Schedule (20 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
5.1	TEDxKinda Project (1 of 5)	CRD-1.C, DAT-2.A, DAT-2.D, DAT-2.E, IOC-1.F	CRD-1.C.1, DAT-2.A.1, DAT-2.A.2, IOC-1.F.11
5.2	Exploring Big Data	DAT-2.E	DAT-2.E.1, DAT-2.E.2, DAT-2.E.3
5.1	TEDxKinda Project (2 of 5)	CRD-1.C, DAT-2.A, DAT-2.D, DAT-2.E, IOC-1.F	CRD-1.C.1, DAT-2.A.1, DAT-2.A.2, IOC-1.F.11
5.3	Big Data Collection	DAT-2.A, DAT-2.C, DAT-2.D	DAT-2.A.1, DAT-2.C.1, DAT-2.C.2, DAT-2.C.3, DAT-2.C.4, DAT-2.C.5, DAT-2.C.6, DAT-2.C.7, DAT-2.C.8, DAT-2.D.1, DAT-2.D.4, DAT-2.D.6
5.4	Digitizing Business Cards	DAT-2.A, DAT-2.C	DAT-2.A.1, DAT-2.C.1, DAT-2.C.2, DAT-2.C.3, DAT-2.C.4, DAT-2.C.5, DAT-2.C.6, DAT-2.C.7, DAT-2.C.8
5.5	Big Data Assessment		
5.6	Spiderbots, Data Persistence, and Data Breaches	DAT-2.D, IOC-2.A	DAT-2.D.1, IOC-2.A.2, IOC-2.A.3, IOC-2.A.4, IOC-2.A.6, IOC-2.A.14
5.7	Big Picture: Privacy vs. Utility	IOC-2.A	IOC-2.A.2, IOC-2.A.3, IOC-2.A.4, IOC-2.A.6, IOC-2.A.14
5.8	Data Mining	DAT-2.E	DAT-2.E.1, DAT-2.E.2, DAT-2.E.3, DAT-2.E.4, DAT-2.E.5
5.9	Association Rule Mining	DAT-2.A, DAT-2.D	DAT-2.A.3, DAT-2.A.4, DAT-2.D.5

5.10	Data Science with Python	DAT-2.D	DAT-2.D.1, DAT-2.D.2
5.1	TEDxKinda Project (3 of 5)	CRD-1.C, DAT-2.A, DAT-2.D, DAT-2.E, IOC-1.F	CRD-1.C.1, DAT-2.A.1, DAT-2.A.2, IOC-1.F.11
5.11	Big Data Reflection	DAT-2.D, IOC-1.E	DAT-2.D.2, IOC-1.E.1, IOC-1.E.2, IOC-1.E.3, IOC-1.E.4, IOC-1.E.5, IOC-1.E.6
5.12	Models and Simulation	AAP-3.F	AAP-3.F.1, AAP-3.F.2, AAP-3.F.3, AAP-3.F.4, AAP-3.F.5, AAP-3.F.6, AAP-3.F.7, AAP-3.F.8
5.1	TEDxKinda Project (4 of 5)	CRD-1.C, DAT-2.A, DAT-2.D, DAT-2.E, IOC-1.F	CRD-1.C.1, DAT-2.A.1, DAT-2.A.2, IOC-1.F.11
5.13	Unit 5 Review	None	None
5.14	Unit 5 Exam	None	None
5.1	Project Presentations (5 of 5)	None	None

Unit 6: Innovative Technologies

As a way of further expanding upon the applications of computer science in the advancement of computational technologies, this unit aims to broaden students' awareness of the computing tools they use and rely on every day and to encourage them to start thinking about the decisions and processes that go into the creation of these technologies.

Students will begin by exploring computing innovations by completing the Explore Curricular Requirement using the College Board activities. Students will then dive into the many key roles that technology plays in their lives, including social networking, search, cloud computing, and the Digital Divide, and examine the ways these ever-evolving technologies have impacted individuals and societies in recent years. With so many of these technologies relying on the internet to connect users and data across varied and remote locations, the students will then "take a peek under the hood" to examine the systems and protocols that make up the global infrastructure of the internet. Finally, students will turn their attention to the past, present, and future of computing to begin imagining the technology that might exist in their future and the role that they might play in bringing it about.

The Big Ideas of Creative Development, Data, Algorithms and Programming, Computing Systems and Networks, and Impact of Computing are all covered in Unit 6. These are the Enduring Understandings covered in this unit:

- CRD-1: Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- CRD-2: Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- DAT-2: Programs can be used to process data, which allows users to discover information and create new knowledge.
- AAP-1: To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- CSN-1: Computer Systems and networks facilitate the transfer of data.
- CSN-2: Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.
- IOC-1: While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.
- IOC-2: The use of computing innovations may involve risks to personal safety and identity.

These are the Computational Thinking Skills addressed:

- 1.A Investigate the situation, context, or task.
- 1.C Explain how collaboration affects the development of a solution.
- 1.D Evaluate solution options.
- 2.B Implement and apply an algorithm.
- 3.A Generalize data sources through variables.
- 4.B Determine the result of code segments.
- 5.A Explain how computing systems work.
- 5.B Explain how knowledge can be generated from data.
- 5.C Describe the impact of a computing innovation.
- 5.D Describe the impact of a gathering data.
- 5.E Evaluate the use of computing based on legal and ethical factors.

Unit 6 Topics

Over the course of this unit and the unit project, the following topics will be covered, and teachers may use these in the AP Classroom to filter questions for formative and summative assessments:

- 1.1 Collaboration
- 1.2 Program Function and Purpose
- 2.3 Extracting Information from Data
- 2.4 Using Programs with Data
- 3.1 Variables and Assignment
- 4.1 The Internet
- 4.2 Fault Tolerance
- 4.3 Parallel and Distributed Computing
- 5.1 Beneficial and Harmful Effects
- 5.2 Digital Divide
- 5.5 Legal and Ethical Concerns
- 5.6 Safe Computing

Explore Curricular Requirement [CR-10]

For assignments 6.2, 6.3, and 6.4, students complete the three activities outlined in the College Board's AP Computer Science Principles Explore Curricular Requirements Teacher Resources:

- Selecting Computing Innovations
- Analyzing Data for Computing Innovations
- Analyzing Impact of Computing

Unit 6 Schedule (23 Days)

Assignment #	Topic	Learning Objectives	Essential Knowledge Statements
6.1	Prototyping the Future Project (1 of 5)	CRD-1.A, CRD-1.C, CRD-2.A	CRD-1.C.1
6.2	Explore Curricular Activity 1	CRD-1.A, CRD-2.A	CRD-A.1.A, CRD-1.A.2, CRD-2.A.1, CRD-2.A.2
6.3	Explore Curricular Activity 2	AAP-1.A, CRD-2.C, CRD-2.D, IOC-2.A	AAP-1.A.3, CRD-2.C.1, CRD-2.C.4, CRD-2.C.6, CRD-2.D.1, IOC-2.A.1, IOC-2.A.5, IOC-2.A.7, IOC-2.A.8, IOC-2.A.9, IOC-2.A.10, IOC-2.A.11, IOC-2.A.12
6.4	Explore Curricular Activity 3	CRD-1.A, CRD-1.B, IOC-1.A, IOC-1.B	CRD-1.A.3, CRD-1.A.5, CRD-1.A.6, CRD-1.B.1, IOC-1.A.1, IOC-1.A.2, IOC-1.A.3, IOC-1.A.4, IOC-1.A.5, IOC-1.B.1, IOC-1.B.2, IOC-1.B.3, IOC-1.B.4, IOC-1.B.5, IOC-1-B.6
6.1	Prototyping the Future Project (2 of 5)	CRD-1.A, CRD-1.C, CRD-2.A	CRD-1.C.1, CRD-1.C.1
6.5	Impact of Computing	DAT-2.B, DAT-2.D, IOC-1.C, IOC-1.F	DAT-2.B.1, DAT-2.B.2, DAT-2.B.3, DAT-2.B.4, DAT-2.B.5, DAT-2.D.3, IOC-1.C.1, IOC-1.C.2, IOC-1.C.3, IOC-1.C.4, IOC-1.C.5, IOC-1.F.10
6.6	The Internet	CSN-1.A, CSN-1.B, CSN-1.C, CSN-1.E	CSN-1.A.1, CSN-1.A.2, CSN-1.A.3, CSN-1.A.4, CSN-1.A.5, CSN-1.A.6, CSN-1.A.7, CSN-1.A.8, CSN-1.B.1, CSN-1.B.2, CSN-1.B.3, CSN-1.B.4,

			CSN-1.B.5, CSN-1.B.6, CSN-1.B.7, CSN-1.E.1, CSN-1.E.2, CSN-1.E.3, CSN-1.E.4, CSN-1.E.5, CSN-1.E.6, CSN-1.E.7
6.7	Internet Protocols	CSN-1.A, CSN-1.B, CSN-1.C	CSN-1.A.5, CSN-1.A.6, CSN-1.B.1, CSN-1.B.2, CSN-1.B.3, CSN-1.B.4, CSN-1.B.5, CSN-1.B.6, CSN-1.B.7, CSN-1.C.1, CSN-1.C.2, CSN-1.C.3, CSN-1.C.4
6.8	World Wide Web	CSN-1.D	CSN-1.D.1, CSN-1.D.2, CSN-1.D.3
6.9	Ethics and Technology	IOC-1.F	IOC-1.F.8, IOC-1.F.9
6.1	Prototyping the Future Project (3 of 5)	CRD-1.A, CRD-1.C, CRD-2.A	CRD-1.C.1
6.10	Sequential, Distributed, and Parallel Computing	CSN-2.A, CSN-2.B, DAT-2.C	CSN-2.A.1, CSN-2.A.2, CSN-2.A.3, CSN-2.A.4, CSN-2.A.5, CSN-2.A.6, CSN-2.A.7, CSN-2.B.1, CSN-2.B.2, CSN-2.B.3, CSN-2.B.4, CSN-2.B.5, DAT-2.C.7
6.11	Cybersecurity	IOC-2.B, IOC-2.C	IOC-2.B.1, IOC-2.B.3, IOC-2.B.4, IOC-2.B.5, IOC-2.B.6, IOC-2.B.7, IOC-2.B.8, IOC-2.B.9, IOC-2.B.10, IOC-2.B.11, IOC-2.C.1, IOC-2.C.2, IOC-2.C.3, IOC-2.C.4, IOC-2.C.5, IOC-2.C.6, IOC-2.C.7
6.1	Prototyping the Future Project (4 of 5)	CRD-1.A, CRD-1.C, CRD-2.A	CRD-1.C.1
6.12	Unit 6 Review	None	None

6.13	Unit 6 Exam	None	None
6.1	Project Presentations (5 of 5)	None	None

Create Performance Task [CR-11]

This section serves to fulfill the Create Performance Task requirements of the AP Computer Science Principles exam. This externally moderated assessment will account for 30% of the student's AP exam score. As such, the work produced in this unit should reflect the sole work of the student and performed in-class with minimal involvement from the classroom teacher. The student may receive collaborative support from a fellow student, but the work submitted should the individual student's own work.

UTeach recommends completing the Create Task after Unit 5. By this point in the curriculum, all the projects, exercises, and classroom discussions from the previous five units will have provided students with extensive, hands-on experience with the exploration, use, and creation of developing programs using the iterative and incremental design process. Students will have learned two different programming languages, Scratch and Python, either of which are acceptable to use for the Create Task. Ultimately, it is up to the teacher as to when to administer the Create Task.

The Create Task Pacing Guide provides more than the 12 hours required of class time with detailed teacher instructions and planning handouts to help teachers guide students through the completion of the Create Task including submission to the Digital Portfolio.

Create Task Schedule (18 days)

Assignment #	Topic	Lessons
CT.1	Create Performance Task: Introduction 12+ hours (approximately 3–4 weeks)	Review instructions and scoring guidelines
CT.2	Review samples	Mock grading of sample student responses
CT.3	Planning	Identify project ideas
CT.4	Develop Scratch program for Create Task	• Develop, implement, and test program • Create PDF of program code • Create video of program • Compile and upload written responses to the digital portfolio
	--------------------OR--------------------	
CT.4	Develop Python program for Create Task	• Develop, implement, and test program • Create PDF of program code • Create video of program • Compile and upload written responses to the digital portfolio

Pedagogical Approaches

The UTeach CS Principles curriculum utilizes Project-Based Learning (PBL) and Culturally Responsive Education (CRE) to better engage students in the learning process. By encouraging students to use critical thinking skills and challenging them to solve authentic and meaningful problems, PBL helps students to develop a deeper and more profound understanding of the power of computation in our everyday lives. This project-based approach is particularly effective in engaging girls and other historically underrepresented groups as well as broadening participation in computing overall. Teachers who are unfamiliar with the goals, methods, and techniques of PBL can learn more at the [Buck Institute for Education](#) website. Likewise, more information about CRE can be found at the [Education Alliance](#) website.

In teaching this curriculum, educators are encouraged to utilize the range of PBL and CRE techniques that have been incorporated into each unit, including *driving questions*, overarching *unit projects*, clear *rubrics*, regular *benchmarks*, *scaffolding activities*, *final products*, and *reflection*. Used together in a coherent, unified manner that actively engages students in the educational process, PBL and CRE strategies can help students improve their retention of learned experiences and develop stronger problem solving, critical thinking, and group communication skills.

Instructional Sequencing

The year-long curriculum consists of six units (each approximately lasting four weeks) and a Python Bootcamp module that have been carefully structured to gently guide novice students through the study of computational technology by first establishing a *context* for the curriculum material, then teaching the *core* skills for creating and using computational tools, followed by demonstrating real-world *applications* of digital technology, and finally allowing the students to *exhibit* the skills they have developed. There is also a unit on the Create Performance Task and an optional unit on Artificial Intelligence.

Driving Questions

Every unit is ultimately guided by one or more driving questions that serve to specify the unifying goal for student inquiry and learning. These questions, which ground each unit in an authentic, real-world context, will be introduced at the start of each unit and then later revisited and reiterated throughout the ensuing instructional modules. Through this regular repetition, teachers can ensure that students always have a clear sense of what they are trying to solve, what they still need to know, and where they stand in terms of achieving their goals for the unit.

Unit Projects

The opening module of each unit also serves as a formal launch of the unit project, an overarching, product-oriented challenge for students to investigate, research, and solve over the course of the unit. The project launch starts with an anchor video that introduces the fundamental problem or challenge to be solved and is intended to spark the students' imaginations and inspire them to want to find a solution. Teachers should then use subsequent classroom discussions, lessons, and activities to help guide students in identifying ways they might approach the project and what they will need to study and learn to develop a complete solution to the stated challenge.

Rubrics

Each unit project is accompanied by a clearly defined rubric that specifies the set of expectations for student work throughout the unit, including an exhaustive list of assessment criteria for the artifacts that students will produce and detailed descriptors for each performance level that a student might demonstrate. Teachers should provide students with the rubric at the start of the unit as part of the initial discussion immediately following the anchor video. Giving the students the rubric at the time of the project launch is critical for setting clear student expectations early in the research and learning process. Over the course of the unit, teachers should regularly refer to the goals and criteria of the rubric to ensure that students remain focused and on pace for meeting the stated requirements.

Benchmarks

Each unit provides the teacher with several benchmark activities, or subtasks, that feed into the larger unit project. Each of these subtasks contributes directly to the final product that the students will create. Teachers can use these benchmarks as intermediate, informal assessments to gauge the progress of each student and/or collaborative group in their mastery of the unit goals.

Scaffolding Activities

The bulk of each unit consists of a series of individual topic lessons, activities, discussions, and hands-on applications that allow the teacher to provide instruction, guidance, and support to students and collaborative groups as they conduct research for the unit project. These scaffolding activities serve to introduce, explain, and encourage the use of the unit's core concepts and skills by providing students with structured opportunities and incentives to explore the material in greater depth. Specifically, *topic lessons* focus on extending existing knowledge of unit concepts through direct instruction and inquiry-based investigations.

Meanwhile the *Unit Project*, *Coding Skills*, and *Big Picture* exercises create opportunities for student-centered, collaborative discussions and exercises that encourage students to explore unit topics from the broader perspectives of creativity, algorithms, and global impact.

Assessments

In addition to informal, formative assessments throughout each unit, student learning and progress is also assessed at the end of each unit through a formal, summative assessment and evaluation of their independent and collaborative efforts on the unit project.

Formal assessments are modeled after the single-select and multiple-select multiple-choice questions of the AP Computer Science Principles exam so that students can familiarize themselves with the scope and style of questions that they can expect to see on the AP exam in May.

Likewise, as preparation for the Create Task that the students will submit to the College Board in May, each student will be required to maintain and document a portfolio of their independent and collaborative work throughout each unit. During the Create Task Unit students are encouraged to draw upon this body of work to produce their final submissions for the College Board.

Final Products and Student Portfolio

At the culmination of each unit, students are expected to present a final product that represents the body of their work and research on the unit project. Through a combination of individual products and collaborative group products, students demonstrate their mastery of the core content and skills for the unit by exhibiting the authentic and purposeful artifacts that they have created. While the exact format of the final product may vary from unit to unit, a key component always includes a public presentation of each student's work before their peers as a way of providing motivation for each student and holding them accountable for their own learning.

In addition, teachers should encourage students to add artifacts of their final product to an ongoing student portfolio that they maintain throughout the course. The contents of this portfolio can later be used as a point of inspiration for the Create Performance Task that the students will need to produce and submit to the College Board as part of the AP Computer Science Principles assessment.

Reflections

Finally, at the end of each unit, teachers should allow their students an opportunity to look back on what they have done, experienced, and learned over the course of the unit and reflect on how their perception of computing may have changed because of these experiences. Similarly, students are expected to discuss how the unit material relates to their own personal interests and to articulate how they could apply their newly learned skills to authentic tasks within their own lives.

Resources and Technical Requirements

UTeach AP Computer Science Principles does not require additional materials to be purchased for implementation.

Websites Accessed Throughout the Curriculum

The websites below will be accessed in this curriculum. You should advise your IT department about these websites prior to beginning the course. Always test site access from your device and the students' devices to ensure a smooth lesson:

> *.codio.com, *. codio.io, uteachcs.github.io, scratch.mit.edu, studio.sketchpad.cc, youtube.com, collegeboard.org, wired.com, collegexpress.com, tiny.cc, wikipedia.org, ars.userfriendly.org, simonsingh.net, csunplugged.org, uscyberpatriot.org, puzzles.com, 20q.net, en.akinator.com, utexas.edu, base64.wutils.com, mathisfun.com, laweekly.com, slate.com, vt.edu, informationisbeautiful.net, tonakai.aki.gs, wfu.edu, jumk.de, vimeo.com, creativecommons.org, bitsbook.com, ted.com, data.gov, ncdc.noaa.gov, knoema.com, geocommons.com, statsilk.com, betterworldflux.com, gapminder.org, wordle.net, textcompactor.com, mapstory.org, heatmaptool.com, mapalist.com, mturk.com, kickstarter.com, fold.it, setiathome.berkeley.edu, starwarsuncut.com, npr.org, maximumedge.com, archive.org, dontbubble.us, duckduckgo.com, niemanlab.org, purplemath.com, whatshouldireadnext.com, forbes.com, communicationnation.blogspot.com, spyrestudios.com, tagxedo.com, apple.com, photopin.com, freeimages.com, openclipart.org, clker.com, piktochart.com, infogr.am, designify.me, duolingo.com, nytimes.com, journalism.org, tools.ietf.org, amazon.com, ieee.org, howtofindmyipaddress.com, ipv6-test.com, info.cern.ch, oracleofbacon.org, berkeley.edu, browserling.com