edhesive

AP Computer Science Principles
Course Syllabus and Planning Guide
(2019-2020)

Curricular Requirements

Curricular Requirements		Page(s)
CR1a	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P1: Connecting Computing.	16, 19, 23, 26, 30, 31
CR1b	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P2: Creating Computational Artifacts.	16, 19, 22, 23, 26, 30, 31
CR1c	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P3: Abstracting.	16, 19, 22, 23, 26, 30, 31
CR1d	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P4: Analyzing Problems and Artifacts.	16, 19, 22, 23, 26, 30, 31
CR1e	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P5: Communicating (both orally and written).	16, 19, 22, 23, 26, 30, 31
CR1f	Students are provided with opportunities to meet learning objectives connected to computational thinking practice P6: Collaborating.	16, 19, 22, 23, 26
CR2a	Students are provided with opportunities to meet learning objectives within Big Idea 1: Creativity.	16, 19, 22, 23, 26, 30, 31
CR2b	Students are provided with opportunities to meet learning objectives within Big Idea 2: Abstraction.	19, 23
CR2c	Students are provided with opportunities to meet learning objectives within Big Idea 3: Data and Information.	22, 26, 30, 31
CR2d	Students are provided with opportunities to meet learning objectives within Big Idea 4: Algorithms.	16, 19, 22, 23
CR2e	Students are provided with opportunities to meet learning objectives within Big Idea 5: Programming.	16, 19, 22, 23, 30, 31
CR2f	Students are provided with opportunities to meet learning objectives within Big Idea 6: The Internet.	16
CR2g	Students are provided with opportunities to meet learning objectives within Big Idea 7: Global Impact.	16, 26, 30, 31
CR3	Students are provided the required amount of class time to complete the AP Through-Course Assessment *Explore – Impact of Computing Innovations* performance task.	6, 10, 23
CR4	Students are provided the required amount of class time to complete the AP Through-Course Assessment *Create – Applications from Ideas* performance task.	6, 10, 31

Introduction

AP Computer Science Principles (AP CSP) is a full-year, rigorous course that introduces students to the foundational concepts of computer science and explores the impact computing and technology have on our society. The course covers a broad range of foundational topics including: programming, algorithms, the Internet, big data, digital privacy and security, and the societal impacts of computing.

About the Course

Edhesive developed this course in partnership with the University of Texas at Austin's UTeach Institute. This custom course combines the esteemed UTeach CS Principles curriculum with additional features and tools specific for a technology-driven student-centered curriculum, including: instructional lesson videos and slides, worked practice problems, unit project scaffolding, student activity and task examples and grading rubrics, enhanced online and offline question banks with College Board-style questions, annotated explanations for all assessment questions, and in-unit mini performance tasks. Additionally, UTeach's lesson plans have been substituted for lesson and unit guides, since they have been revised to focus less on teacher-driven directives for students ("say this," "do this," etc.) and more on teaching tips and strategies.

All schools using Edhesive's AP CSP course should use this syllabus.

Course Overview

Prerequisites
The College Board suggests students successfully complete a first year, high school Algebra course prior to enrolling in AP CSP. An Algebra course will provide a strong foundation in problem solving, basic linear functions, composition of functions, and the Cartesian (x,y) coordinate system. These skills and topics are essential for student facility in this course. For further preparation, we recommend students complete our Introduction to Computer Science course prior to taking this course. That course introduces students to fundamentals of computing, providing a foundation on which this course can build.

The College Board adheres to an open enrollment policy for this course, meaning any student that is willing and academically prepared can participate in the course.

Pedagogical Approach
Edhesive's AP CS Principles course follows the blended learning model. It takes a student-centered approach powered by technology to help realize the goal of high achievement for all students. The course promotes student engagement, independent thought and interactive collaboration with peers. Student-centric lessons, activities and assessments are paired with augmentative teacher-centric lesson, activity and task guides and reporting to empower teachers to empower students. Additionally, teacher and student forums with moderation and input from Edhesive staff and team of teaching assistants provide dynamic community and support.

Programming Requirements
The coding languages Scratch and Processing are both used in this course. Scratch is a free block-based programming environment that is accessible enough for beginners, yet can support the development of advanced algorithms used in more complex games and applications. Processing is a text-based language with syntax similar to Java; it also uses a free widely available programming environment.

Course Goals

Edhesive's AP CSP course fully addresses the College Board's AP Computer Science Principles Curriculum Framework. The framework defines two through-course curricular requirements: six "computational thinking practices" and seven "big ideas." Additionally, the framework describes in detail what students should be able to do, know, and retain by the end of the course with three types of expressions: Enduring Understandings, Learning Objectives, and Essential Knowledge Statements. A basic overview of each of these items is provided below, and we encourage instructors to read more about them in the AP Computer Science Principles Curriculum Framework.

Six Computational Thinking Practices
The six Computational Thinking Practices demonstrate important aspects of the work computer scientists engage in.

Computational Thinking Practices					
P1	P2	P3	P4	P5	P6
Connecting Computing	Creating Computational Artifacts	Abstracting	Analyzing Problems and Artifacts	Communicating	Collaborating

Seven Big Ideas
The course material focuses on Seven Big Ideas. These ideas encompass concepts that are foundational to computer science.

Big Ideas	
Creativity [Big Idea 1] Abstraction [Big Idea 2] Data and Information [Big Idea 3] Algorithms [Big Idea 4]	Programming [Big Idea 5] The Internet [Big Idea 6] Global Impact [Big Idea 7]

Enduring Understandings
Enduring Understandings (EUs) describe the concepts students should ultimately retain from this course. The goal is for learning objectives and essential knowledge statements to build enduring understandings.

Learning Objectives
Learning objectives (LOs) articulate what students are expected to know by the end of the course. Each learning objective corresponds to one of the Seven Big Ideas and one or more computational thinking practices. Both the multiple choice exam and through-course performance tasks test students' mastery of these learning objectives.

Essential Knowledge Statements
Essential knowledge statements (EKs) provide facts or concepts students should be familiar with to prove their understanding of the learning objectives.

The AP Exam

The AP Exam will test students on their understanding of the seven big ideas through a multiple-choice exam and two through-course performance tasks. Together, these components will be used to calculate the AP score (on a 1-5 scale).

Multiple Choice Exam
The 74-question multiple choice exam will test students' understanding of computational logic, which they will learn over the course of the year. This section is programming language agnostic, meaning they don't have to know a formal coding language to complete this part of the exam. The multiple-choice exam will be on May 15, 2020, and accounts for 60% of a student’s total AP score.

Performance Tasks
The two performance tasks are called the Create Task and the Explore Task. These tasks function as projects that students must complete independently and submit online prior to taking the multiple-choice portion of the exam. These two tasks account for 40% of a student’s total AP score. Broken down between the two tasks, the Create Task is worth 24% of a student’s overall score and the Explore Task is worth 16% of a student’s overall score.

Create Task
On the Create Task, students will create their own program. Students will submit a video of their program running and a written response describing how their program works. The Create Task accounts for 24% of the overall AP score, and students must be given a minimum of 12 hours in class to work on it.

Explore Task
On the Explore Task, students will identify a computing innovation, learn how it works, and explore the impact it has on society. They will submit a digital artifact (video, art, etc.) and written response explaining their research findings. The explore task accounts for 16% of the overall AP score, and students will have a minimum of 8 hours in class to work on it.

Students are required to submit both performance tasks via the College Board’s online Digital Portfolio by April 30, 2020 at 11:59 p.m.

You can read the official overview of the AP CSP Assessment on pages 1-2 in the College Board's AP CSP Assessment Overview and Performance Task Directions for Students document.

Course Materials

Edhesive's AP Computer Science Principles has an introductory unit, six instructional units, two mini modules, and a final AP review unit. Each is strategically designed to prepare students for the AP CSP Exam. The course units consist of daily lessons, instructional videos, lesson slides, lesson activities, code-along exercises, projects, vocabulary reviews, AP test preparation, quizzes, and tests.

Student Lessons

The student lessons are typically composed of the following components:

- Introduction: a high-level overview of the lesson.
- Objectives: a list of what students will learn and do during the lesson.
- Instructional Video(s): one or more explanatory or demo videos taught by an expert computer science teacher, some of which include code-along activities. Most videos are accompanied by corresponding downloadable slides for review or notetaking.
- In-Lesson Activities: these activities take place during the lesson, prior to the graded exercises, such as a class discussion, interactive, or code-along activities.
- Summary: a text version of the key concepts in the lesson.
- Vocabulary: a list of terms and definitions for the lesson.
- Lesson Exercises: one or more graded exercises that ask students to apply or extend the concepts in the lesson. Lesson exercises include coding activities, discussions, research, AP-style question practices (such as pseudocode or number conversions), strategic games, computational practices (non-coding), and more. The variety of formats and tasks prepare students for the diverse questions and tasks on the AP CSP Exam.

Other Assignments

In addition to the daily lessons and exercises, the AP CSP curriculum also offers other types of assignments for students.

- Vocabulary Practices: Each unit has a vocabulary practice that helps students to reinforce the unit's keywords. These practices are game-based and allow for several different types of practice formats like matching and flashcards. The vocabulary practices are not a graded assignment.
- Big Picture Exercises: classroom investigations or discussions that examine the cultural and societal impact of emerging technologies.
- Unit Projects: Four of the six units include an extended project that challenges the students to apply various concepts from the current and past units in a new or more complex way. These projects demand a high level of critical thinking and problem solving.
- Mini-Performance Tasks: The course includes two mini-performance tasks that help prepare students for the official AP CSP Create and Explore Tasks. These are multi-day projects that mirror the types of tasks and activities that students must complete to meet the College Board requirements for the official tasks.

Assessments

The AP Computer Science Principles course offers two types of assessments: quizzes and tests.

- Quizzes: Each unit has two short quizzes that act as a checkpoint for understanding. These quizzes range from 4-7 multiple choice questions with shuffled answers.
- Tests: Each unit has a summative test at its conclusion. The tests are always 20 multiple choice questions with shuffled answers.

Grading

For Edhesive's AP CSP course, there are several types of formative and summative assessments, all intended to prepare students for the end-of-year 74-question multiple choice exam (60% of the overall AP score) as well as the through-course Create and Explore performance tasks (40% of the overall AP score).

The default course grading scheme maps to this 60/40 breakdown:

- 10% Unit Quizzes
- 50% Unit Exams
- 10% Lesson Exercises and Activities
- 10% Mini Performance Tasks
- 20% Unit Projects

Teacher Sidebar

The Teacher Sidebar, which is located within a teacher's version of the student course, contains several types of resources that will help you in facilitating the course, including:

- Lesson guides that detail the lesson objectives, lesson components, as well as indicators of key points to emphasize and common misconceptions
- Supplemental resources, such as worksheets or unit project rubrics
- Alternative assessments, such as paper-based and alternate versions of quizzes and tests for each unit
- Answer keys, including annotated solutions to quizzes and tests

Course Sequencing

The year-long curriculum directly addresses the College Board's AP Computer Science Principles curriculum framework. It has been designed carefully to teach students the core skills for 1) creating and using computational tools 2) applying logical reasoning and creative problem solving and 3) recognizing real-world applications for digital technology. As described above, it is comprised of an introductory unit, six instructional units, two mini modules for performance task preparation, and a final unit focused on the course's AP exam. The curriculum also provides two windows of time for students to complete the required Create and Explore Tasks.

The sequencing and a high-level description of all components is outlined below:

Content Overview		
Unit 0: Course Introduction Enter the world of computer science by learning about the field itself and the goals of this AP-level course.	**Big Ideas:** Creativity [1] Algorithms [4] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5 **Enduring Understandings (EUs):** 1.2, 4.1, 5.1, 5.2, 7.2
Unit 1: Computational Thinking Study the iterative development process, and start applying it to build your own programs in *Scratch.*	**Big Ideas:** Creativity [1] Abstraction [2] Algorithms [4] Programming [5]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5 **Enduring Understandings (EUs):** 1.1, 1.2, 2.2, 4.1, 5.1, 5.2, 5.4, 5.5
Unit 2: Programming Examine computational logic structures and problem solving capabilities for programs in text-based algorithms, AP-style Pseudocode, and *Scratch*.	**Big Ideas:** Creativity [1] Abstraction [2] Algorithms [4] Programming [5] The Internet [6] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** 1.1, 1.2, 2.1, 2.2, 2.3, 4.1, 4.2, 5.1, 5.2, 5.3, 5.4, 5.5, 6.3, 7.2
Unit 3: Data Representation Explore the different means of representing information digitally.	**Big Ideas:** Creativity [1] Abstraction [2] Data and Information [3] Algorithms [4] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** 1.2, 2.1, 2.2, 2.3, 3.3, 4.1, 4.2, 5.1, 5.3, 5.4, 5.5, 7.3
Mini Create Task Module Learn about the Create Performance Task component of the AP exam, and practice the skills required for it.	**Big Ideas:** Creativity [1] Abstraction [2] Algorithms [4] Programming [5]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** 2.2, 4.1, 5.1, 5.2, 5.3, 5.4, 5.5

Unit 4: Digital Media Processing Use Processing to programmatically manipulate digital images and audio.	**Big Ideas:** Creativity [1] Abstraction [2] Data and Information [3] Algorithms [4] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTP):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** 1.1, 1.2, 1.3, 2.1, 2.2, 3.1, 3.3, 4.1, 5.1, 5.2, 5.3, 5.4, 5.5, 7.2, 7.3
Create Performance Task Students demonstrate their learning by creating a portfolio of their work for submission to the College Board.	**Big Ideas:** Abstraction [2] Algorithms [4] Programming [5]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P5, P6 **Enduring Understandings (EUs):** 2.2, 4.1, 5.1, 5.2, 5.3, 5.4, 5.5
Unit 5: Big Data Discover new knowledge through the use of large data sets.	**Big Ideas:** Creativity [1] Abstraction [2] Data and Information [3] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** 1.2, 2.3, 3.1, 3.2, 3.3, 5.1, 7.1, 7.2, 7.3, 7.5
Unit 6: Innovative Technologies Assess the current state of technology and investigate its role in our everyday lives.	**Big Ideas:** Abstractions [2] Data and Information [3] The Internet [6] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P3, P4, P5 **Enduring Understandings (EUs):** 2.2, 3.2, 3.3, 5.3, 6.1, 6.2, 6.3, 7.1, 7.2, 7.3, 7.4
Mini Explore Task Module Learn about the Explore Performance Task component of the AP exam, and practice the skills required for it.	**Big Ideas:** Creativity [1] Data and Information [3] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5 **Enduring Understandings (EUs):** 1.2, 3.3, 5.2, 5.4, 7.1, 7.2, 7.3, 7.4, 7.5
Explore Performance Task Students demonstrate their learning by creating a portfolio of their work for submission to the College Board.	**Big Ideas:** Creativity [1] Data and Information [3] Programming [5] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5 **Enduring Understandings (EUs):** 1.2, 3.3, 5.2, 5.4, 7.1, 7.2, 7.3, 7.4, 7.5
Unit 7: The AP Exam Students review and prepare for all components of the AP Exam.	**Big Ideas:** Creativity [1] Abstraction [2] Data and Information [3] Algorithms [4] Programming [5] The Internet [6] Global Impact [7]	**Computational Thinking Practices (CTPs):** P1, P2, P3, P4, P5, P6 **Enduring Understandings (EUs):** N/A

Unit 0: Course Introduction

Unit 0 is an introduction to the AP Computer Science Principles course. The unit exposes students to the foundational topics of computer science and computing. Additionally, it introduces the major topics and components on the AP exam, so students will become familiar with the big ideas and computational thinking practices around which the course is focused. Before finishing the unit, students will engage with their preconceived notions about computer science and challenge these ideas.

Unit 0 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Course Context	Welcome to AP Computer Science Principles	None	None
	Computer Science Fundamentals	1.2, 4.1, 5.1, 5.2, 7.2	1.2.1, 1.2.5, 4.1.2, 5.1.1, 5.2.1, 7.2.1
	Course Structure	None	None
Course Resources	Meet the Virtual Instructor	None	None
	Student Forum	None	None
	Forum Guidelines	None	None
	Honor Code	None	None
Self-Evaluation	Entry Questionnaire	None	None

Unit 0 Topics

Course Context [EU 1.2, EU 4.1, EU 5.1, EU 5.2, EU 7.2] [LO 1.2.1, LO 1.2.5, LO 4.1.2, LO 5.1.1, LO 5.2.1, 7.2.1] (P1, P2, P3, P4, P5)

- Students will examine and discuss the motivations behind a number of high-profile individuals in the field of programming.
- Students will discuss the benefits of programming as a tool and a profession.
- Students will discuss the impact computing has on society, business, and the economy.
- Students will examine the ideas of computational thinking and computational artifacts.

Course Resources

- Students will become familiar with the resources on the Edhesive platform.

Self-Evaluation

- Students will consider their relationship with computer science and programming.

Unit 1: Computational Thinking

This unit lays the foundation for computational logic. Students first explore the iterative development process, seeing how an idea translates to a real, functioning program. Then, they take a closer look at this process by examining algorithms, languages, program execution, and the through-course concept of abstraction. For the second half of the unit, students get started coding in Scratch. Using this visual, block-based programming language, they learn basic programming concepts and constructs, including user input and variables. In creating programs of their own, they have the opportunity to apply the iterative development process. Over the course of the unit, students learn how to build computational artifacts and solve computational problems - two skills essential to the rest of the course.

There are no major projects in this unit, but there are several post-lesson opportunities for students to apply the iterative development process and basic programming concepts.

Unit 1 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Program Development	The Iterative Development Process	5.1	5.1.2
	Algorithms	4.1, 5.2	4.1.1, 4.1.2, 5.2.1
	Languages	2.2, 4.1	2.2.3, 4.1.2
	Idea to Execution	2.2, 4.1, 5.2	2.2.3, 4.1.1, 4.1.2, 5.2.1
Visual Programming	Getting Started in Scratch	1.1, 5.1	1.1.1, 5.1.2
	Programming with Blocks	1.2, 5.1, 5.2	1.2.1, 5.1.1, 5.1.2, 5.2.1
Program State	User Input and Storage	1.2	1.2.1
	Defining Variables	5.2	5.2.1
	Applying Variables	5.2, 5.4, 5.5	5.2.1, 5.4.1, 5.5.1

Unit 1 Topics

Program Development [EU 2.2, 4.1, 5.1, 5.2] [LO 2.2.3, LO 4.1.1, LO 4.1.2, LO 5.1.2, LO 5.2.1] (P2, P3, P5)

- Students will examine strategies for approaching large-scale problems.
- Students will explore the non-linear approach to solving problems with the iterative development process.
- Students will identify a number of common features of algorithms, including sequencing, selection, and repetition.
- Students will design and evaluate text-based algorithms.
- Students will examine the need for clarity and precision in communicating an algorithmic solution to a problem.
- Students will examine the shortcomings and ambiguities of natural languages.
- Students will identify the elements of clear communication, including well-specified grammar, vocabulary, and syntax.
- Students will analyze the need for artificial programming languages.
- Students will compare high-level languages with low-level languages.
- Students will examine the process in which a program is written in a high-level language, compiled into a low-level language, loaded into memory, and then executed by a processor.

Visual Programming [EU 1.1, EU 1.2, EU 5.1, EU 5.2] [LO 1.1.1, 1.2.1, LO 5.1.1, LO 5.1.2, LO 5.2.1] (P2, P3)

- Students will utilize a graphical editor to read, construct, and execute dynamic programs.
- Students will examine, modify, and execute programs developed by others.
- Students will examine how well-specified behavior of objects can be constructed through sequential actions and operations.
- Students will examine a number of common programming errors.
- Students will explore a number of common debugging strategies.
- Students will develop solutions for correcting common programming errors.

Program State [EU 1.2, EU 5.2, EU 5.4, EU 5.5] [LO 1.2.1, LO 5.2.1, LO 5.4.1, LO 5.5.1] (P1, P2, P3, P4)

- Students will write programs that incorporate dynamic, user-driven, keyboard controls and input.
- Students will examine how the dynamic state of an object or program can be stored and changed using variables.
- Students will analyze the role of clear, descriptive names for objects, behaviors, variables, and other identifiers in maintaining the readability of code.
- Students will analyze and evaluate the correctness of their programs.

Unit 2: Programming

This unit focuses on the three main control structures utilized within algorithms and programs: sequencing, selection, and iteration. Students first examine these structures conceptually, and then learn how to formally construct and evaluate them in Scratch and AP-style Pseudocode. In doing so, they hone their programming abilities and become familiar with the importance of precise commands and well-structured logic. Building on this knowledge, students explore how abstraction can be applied to algorithmic solutions using procedures, and examine 1) how algorithmic solutions should be efficient and help programs scale 2) what happens when a problem is not able to be solved with an algorithm. At the end of the unit, students get a glimpse of how design documentation for hardware components employs computational logic and abstraction just like programming.

There is one major project in this unit: the Password Generator Project.

Unit 2 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Control Structures	Defining Sequencing	4.1, 5.2	4.1.1, 4.1.2, 5.2.1
	Applying Sequencing	4.1, 5.1, 5.2, 5.4	4.1.1, 4.1.2, 5.1.2, 5.2.1, 5.4.1
Coding Skills	Pseudocode	4.1	4.1.2
Control Structures	Defining Selection	4.1, 5.5	4.1.1, 4.1.2, 5.5.1
	Applying Selection	1.1, 1.2, 4.1, 5.1, 5.2, 5.4, 5.5	1.1.1, 1.2.1, 1.2.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.2.1, 5.4.1, 5.5.1
	Defining Iteration	4.1	4.1.1, 4.1.2
	Applying Iteration	1.1, 1.2, 4.1, 5.1, 5.2, 5.4	1.1.1, 1.2.1, 1.2.5, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.2.1, 5.4.1
Procedural Abstraction	Procedures	2.1, 2.2, 5.3	2.1.1, 2.2.2, 5.3.1
Decidability and Efficiency	Solvability & Performance	4.2, 5.1, 5.2	4.2.1, 4.2.2, 4.2.3, 4.2.4, 5.1.2, 5.2.1
Big Picture	Moore’s Law	5.2, 7.2	5.2.1, 7.2.1

Hardware Abstraction	Logic Gates & Hardware	2.1, 2.2, 2.3	2.1.1, 2.1.2, 2.2.3, 2.3.1, 2.3.2
Unit Project	Password Generator Project	1.1, 1.2, 4.1, 4.2, 5.1, 5.2, 5.4, 6.3, 7.2	1.1.1, 1.2.1, 1.2.4, 1.2.5, 4.1.1, 4.1.2, 4.2.1, 4.2.4, 5.1.2, 5.1.3, 5.2.1, 5.4.1, 6.3.1, 7.2.1

Unit 2 Topics

Control Structures [EU 1.1, EU 1.2, EU 4.1, EU 5.1, EU 5.2, EU 5.4, EU 5.5] [LO 1.1.1, LO 1.2.1, LO 1.2.2, LO 1.2.5, LO 4.1.1, LO 4.1.2, LO 5.1.1, LO 5.1.2, LO 5.2.1, LO 5.4.1, LO 5.5.1] (P1, P2, P3, P4, P5)

- Students will examine a number of common features of algorithms, including sequencing, selection, and repetition.
- Students will examine how well-specified behavior of objects can be constructed through sequential actions and operations.
- Students will examine the uses of selection statements in programming.
- Students will analyze the differences between simple selection and complex, nested selection statements.
- Students will examine the use of the Boolean operators "AND," "OR," and "NOT" in constructing complex conditional statements.
- Students will examine the uses of iteration statements in programming.
- Students will consider how to make a sequence of events more efficient with iteration statements.
- Students will combine sequencing, selection, and repetition structures alongside programming constructs like user input and variables to create computational artifacts.

Coding Skills [EU 4.1] [LO 4.1.2] (P5)

- Students will examine how pseudocode can outline algorithmic processes.
- Students will read, execute, and construct algorithms in AP-style pseudocode.

Procedural Abstraction [EU 2.1, EU 2.2, EU 5.3] [LO 2.1.1, LO 2.2.2, LO 5.3.1] (P3)

- Students will compare the methods and relative efficiencies of different algorithms.

Decidability and Efficiency [EU 4.2, EU 5.1, EU 5.2] [LO 4.2.1, LO 4.2.2, LO 4.2.3, LO 4.2.4, LO 5.1.2, LO 5.2.1] (P1, P2, P3, P4)

- Students will examine the factors that affect the decidability of a problem.
- Students will identify which problems can and cannot always be solved by an algorithm.
- Students will examine methods of comparing equivalent algorithms for relative efficiency.
- Students will evaluate the relative efficiency of equivalent algorithms.
- Students will identify factors that allow solutions to scale efficiently.

Big Picture [EU 5.2, EU 7.2] [LO 5.2.1, LO 7.2.1] (P1, P3)

- Students will examine the implications of Moore's Law on the research and development of new and existing technologies.

Hardware Abstraction [EU 2.1, EU 2.2, EU 2.3] [LO 2.1.1, LO 2.1.2, LO 2.2.3, LO 2.3.1, LO 2.3.2] (P3, P5)

- Students will explore the logical processes implement in hardware design documentation.

Unit Project [EU 1.1, EU 1.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 6.3, EU 7.2]
The Password Generator Project occurs after all lesson components, and is a collaborative, in-class activity. In this Scratch programming project, students will explore data security considerations and develop a program for generating unique, secure passwords. Students will:

- Design an algorithm for generating a custom, reproducible password that is uniquely different for each website (e.g., using the domain name as a seed, etc.).
- Write pseudocode to describe each step of the algorithm used to generate a password.
- Exchange algorithms with peers and share feedback with each other on the clarity of the pseudocode and the strengths and weaknesses of the algorithms.
- Construct trace tables documenting the result of each step of the algorithm in generating passwords for different domains.
- Design code in Scratch to implement the password-generating algorithm.

This project encompasses the following College Board curricular requirements: LO 1.1.1 [P2], LO 1.2.1 [P2], LO 1.2.4 [P6], LO 1.2.5 [P4], LO 4.1.1 [P2], LO 4.1.2 [P5], LO 4.2.1 [P1], LO 4.2.4 [P4], LO 5.1.2 [P2], LO 5.1.3 [P6], LO 5.2.1 [P3], LO 5.4.1 [P4], LO 6.3.1 [P1], LO 7.2.1 [P1].

Unit 3: Data Representation

In this unit, students explore the different ways digital information can be of represented, stored, and manipulated on a computer. They look at the various levels of abstraction that are used in the digital representation of discrete data and information. Initially, students will 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. Next, they will examine the distinctions between analog and digital forms of representation. Finally, students will learn about lists, a common abstract data type that can be utilized in programs. They will explore the characteristics of lists and how they can be used to search and sort data.

There is one major project in this unit: the Unintend'o Project.

Unit 3 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Binary Encoding of Information	Binary	2.1	2.1.1, 2.1.2
	Base Conversions	2.1	2.1.1
	ASCII vs. Unicode	1.2, 2.1, 3.3	1.2.5, 2.1.1, 2.1.2, 3.3.1
Coding Skills	Programming Binary	1.2, 2.1, 5.3, 5.5	1.2.2, 1.2.3, 2.1.1, 2.1.2, 5.3.1, 5.5.1
Digital Approximations	Digitization	2.1, 2.2	2.1.2, 2.2.1
	Analog vs. Digital Data	2.2, 5.3, 7.3	2.2.1, 5.3.1, 7.3.1
Big Picture	Reselling Digital Music	7.3	7.3.1
Lists	Making a List	5.3, 5.5	5.3.1, 5.5.1
	Processing a List	4.1, 5.3, 5.5	4.1.1, 5.3.1, 5.5.1
	Sorting a List	4.2, 5.1, 5.3, 5.4	4.2.4, 5.1.2, 5.3.1, 5.4.1
	Lists in Pseudocode	4.1	4.1.2
Unit Project	Unintend'o Project	2.1, 2.2, 2.3, 4.1, 5.1, 5.3, 5.5	2.1.1, 2.1.2, 2.2.1, 2.2.3, 2.3.1, 4.1.1, 4.1.2, 5.1.2, 5.1.3, 5.3.1, 5.5.1

Unit 3 Topics

Binary Encoding of Information [EU 1.2, EU 2.1, EU 3.3] [LO 1.2.5, LO 2.1.1, LO 2.1.2, LO 3.3.1] (P3, P4, P5)

- Students will examine how numerical values are represented using different bases, including decimal and binary.
- Students will explore methods of converting values from decimal to binary and binary to decimal.
- Students will examine the exponential relationship between the number of digits and their range of representable values.
- Students will examine how alphanumeric characters and symbols may be represented using ASCII and Unicode character mappings.
- Students will analyze the differences in state space between ASCII and Unicode standards.
- Students will explore how the interpretation of binary data is dependent upon its intended format and use, including base-64, bitmaps (*.BMP), plaintext (*.TXT), audio (*.MP3), etc

Coding Skills [EU 1.2, EU 2.1, EU 5.3, EU 5.5] [LO 1.2.2, LO 1.2.3, LO 2.1.1, LO 2.1.2, LO 5.3.1, LO 5.5.1] (P1, P2, P3, P5)

- Students will construct a Scratch program that simulates candles on a birthday cake being lit so as to show the user's age in binary.

Digital Approximations [EU 2.1, EU 2.2, EU 5.3, EU 7.3] [LO 2.1.2, LO 2.2.1, LO 5.3.1, LO 7.3.1] (P2, P3, P4, P5)

- Students will examine the implications of variable-width encodings (e.g., Morse code) versus fixed-width encodings (e.g., Baudot code).
- Students will explore ways in which natural phenomena may be represented digitally.
- Students will analyze the extent to which digital approximations accurately reflect the reality that they represent.
- Students will analyze the differences between discrete (digital) and continuous (analog) representations of natural phenomena.
- Students will examine the social implications of the ease with which perfect digital copies can be made.

Big Picture [EU 7.3] [LO 7.3.1] (P4)

- Students will examine and discuss the legality of reselling "used" digital music.

Lists [EU 4.1, EU 4.2, EU 5.1, EU 5.3, EU 5.4, EU 5.5] [LO 4.1.1, LO 4.1.2, LO 4.2.4, LO 5.1.2, LO 5.3.1, LO 5.4.1, LO 5.5.1] (P1, P2, P3, P4, P5)

- Students will examine the use of lists as ordered data structures that may contain multiple values.
- Students will investigate the use of index values to represent the position of an item in a list.
- Students will analyze the implications of accessing an index position beyond the bounds of a list.
- Students will investigate common operations for processing elements of a list, including searching for an element, removing an element, swapping the positions of two elements, or sorting an entire list into ascending or descending order.
- Students will examine the implications of case-sensitivity on ordered lists of strings.
- Students will consider how lists can appear in pseudocode.

Unit Project [EU 2.1, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.3, EU 5.5]
The Unintend'o Project is a collaborative, culminating activity positioned at the end of the unit. In this Scratch programming project, students will write a program that directs the input of a video game controller. It exposes how bits and binary can work to turn on and off functionalities within programs. Students will:

- Map each of six controls (UP, DOWN, LEFT, RIGHT, A, and B) to individual bits.
- Map each binary pattern of button presses to different game actions (e.g., walk forward, walk backward, turn left, turn right, jump, duck, whirl, leap, crawl, etc.).
- Use a list to track the history of button presses.
- Write detailed specifications and justifications for each button-to-action mapping of your design.
- Collaborate with peers throughout the design and development process to determine end-user requests for features and to share feedback on design and implementation strategies.
- Write documentation detailing the use of the program and its features using appropriate terminology.
- Develop a Scratch program that acts as a device driver for a video game controller interface.

This project encompasses the following College Board curricular requirements: LO 2.1.1 [P3], LO 2.1.2 [P5], LO 2.2.1 [P2], LO 2.2.3 [P3], LO 2.3.1 [P3], LO 4.1.1 [P2], LO 4.1.2 [P5], LO 5.1.2 [P2], LO 5.1.3 [P6], LO 5.3.1 [P3], LO 5.5.1 [P1].

Mini Create Task Module

This mini-performance task module is a multi-day activity that gives students a chance to deepen their understanding of the AP CSP Create Task. They begin by exploring the requirements of the task itself. Then, they move on to evaluate sample student submissions against the official College Board rubric. After this, students work on a mock create task, learning how to fulfill the project requirements themselves. Students will:

- begin designing their own program using the iterative development process.
- record a video of the program running.
- provide a written explanation of the purpose, process, algorithms, and abstractions in their design.
- practice submitting their work in the format College Board requires.
- review a peer's work and provide feedback, based on the official Create Task rubric.

This project encompasses the following College Board curricular requirements: [EU 1.2, EU 2.2, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5] [[LO 1.2.1 [P2], LO 1.2.2 [P2], LO 1.2.3 [P2], LO 1.2.4 [P6], LO 2.2.1 [P2], LO 2.2.2 [P3], LO 4.1.2 [P5], LO 5.1.1 [P2], LO 5.1.2 [P2], LO 5.1.3 [P6], LO 5.2.1 [P3], 5.3.1 [P3], 5.4.1 [P4], 5.5.1 [P1]].

Unit 4: Digital Media Processing

In Unit 4, students will use Processing to programmatically manipulate digital images and audio. The unit starts by guiding students through the transition of programming in Processing, which is a high-level, procedural, text-based language. In Processing, students will explore the characteristics of the RGB color model and its use in encoding digital images. They will also investigate the methods of representing and modifying digital audio, including Auto-Tune and audio compression. The unit concludes with a summary of the compression methods related to digital media processing.

There is one major project in this unit: the Image Filter Project.

Unit 4 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Procedural Programming	Introduction to Processing	1.2, 2.2, 4.1, 5.3	1.2.2, 2.2.1, 2.2.2, 4.1.2, 5.3.1
	Scratch vs. Processing	4.1, 5.4	4.1.1, 5.4.1
	Drawing	5.1, 5.3, 5.4	5.1.2, 5.3.1, 5.4.1
	Mouse Input	5.2	5.2.1
	Keyboard Input and Loops	4.1, 5.5	4.1.1, 5.5.1
Image Manipulation	RGB Color	2.1	2.1.1, 2.1.2
	Hexadecimal	2.1	2.1.1
	Raster Images	1.2, 1.3, 3.1, 4.1, 5.3	1.2.3, 1.3.1, 3.1.1, 4.1.1, 5.3.1
	Encoding Schemes	2.1, 3.3	2.1.1, 3.3.1
	Digital Manipulation	1.1, 1.2, 1.3	1.1.1, 1.2.1, 1.3.1
Big Picture	Ethics of Digital Manipulation	7.3	7.3.1
	Intellectual Property	7.2, 7.3	7.2.1, 7.3.1

Audio Manipulation	Digital Audio	1.3, 2.1	1.3.1, 2.1.1
	Audio Processing	1.2, 1.3, 2.1, 7.3	1.2.3, 1.3.1, 2.1.2, 7.3.1
Digital Media Compression	Compression Algorithms	3.3	3.3.1
Unit Project	Image Filter Project	1.2, 1.3, 3.3, 4.1, 5.1, 5.2, 5.3, 5.4	1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.3.1, 3.3.1, 4.1.1, 4.1.2, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1

Unit 4 Topics

Procedural Programming [EU 1.2, EU 2.2, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5] [LO 1.2.2, LO 2.2.1, LO 2.2.2, LO 4.1.1, LO 4.1.2, LO 5.1.2, LO 5.2.1, LO 5.3.1, LO 5.4.1, LO 5.5.1] (P1, P2, P3, P4, P5)

- Students will explore the capabilities of a text-based programming language (Processing).
- Students will compare and contrast the programming capabilities of a visual programming language (Scratch) with those of a text-based programming language (Processing).
- Students will write programs that make use of parameterized methods to invoke specific behaviors.
- Students will understand the importance of using proper punctuation and syntax when coding in a text-based programming language.
- Students will use event handlers to animate on-screen effects and respond to mouse and keyboard input.
- Students will write code using common programming constructs like conditional if() for selection and while() loops for iteration.

Image Manipulation [EU 1.1, EU 1.2, EU 1.3, EU 2.1, EU 3.1, EU 3.3, EU 4.1, EU 5.3] [LO 1.1.1, LO 1.2.1, LO 1.2.3, LO 1.3.1, LO 2.1.1, LO 2.1.2, LO 3.1.1, LO 3.3.1, LO 4.1.1, LO 5.3.1] (P2, P3, P4, P5)

- Students will examine the structure of raster images as compositions of individual pixels.
- Students will explore various methods of representing color, including RGB, CMYK, and HSV.
- Students will explore the various colors that can be produced by the combination of different ratios of red, green, and blue light.
- Students will perform base conversions for decimal, binary, and hexadecimal number systems.
- Students will modify the color channels of pixels in an image to produce a variety of effects.
- Students will design algorithms for modifying the pixels in an image in prescribed ways to create custom image filters.
- Students will explore the difference between lossy and lossless encoding schemes of several common image file formats.

Big Picture [EU 7.2, EU 7.3] [LO 7.2.1, LO 7.3.1] (P1, P4)

- Students will explore the positive and negative consequences of digitally altering images.
- Students will discuss the ethics of digitally manipulating images, especially in the context of journalism.
- Students will discuss the issues related to intellectual property.
- Students will explore the limitations and rights associated with a number of common licenses, including Creative Commons.

Audio Manipulation [EU 1.2, EU 1.3, EU 2.1, EU 7.3] [LO 1.2.3, LO 1.3.1, LO 2.1.1, LO 2.1.2, LO 7.3.1] (P2, P3, P4, P5)

- Students will analyze the differences between analog and digital sound.
- Students will explore the roles that sampling rate and bit depth play in determining the quality of digitized sound.
- Students will explore methods of programmatically generating digital audio.
- Students will explore methods of programmatically altering and modifying digital audio by adjusting volume, pitch, and sampling rate.
- Students will explore the methods and effects of compression algorithms in reducing the amount of data needed to represent an audio sample.

Compression Algorithms [EU 3.3] [LO 3.3.1] (P4)

- Students will revisit the concept of lossy and lossless compression for digital media.

Unit Project [EU 1.2, EU 1.3, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4]
The Image Filter Project is an in-class, collaborative activity that occurs at the end of the unit. In this Processing programming project, students will use their text-based programming skills to develop a program that manipulates digital images similarly to a filter in a photo app. Students will:

- design and implement a program for filtering digital images.
- develop code to systematically transform an image by mathematically manipulating its bits, pixel by pixel.
- write documentation detailing the use of the program and its features using appropriate terminology.
- explain the design and implementation choices by demonstrating and sharing the finished programs with peers.

This project encompasses the following College Board curricular requirements: LO 1.2.1 [P2], LO 1.2.2 [P2], 1.2.3 [P2], 1.2.4 [P6], 1.3.1[P2], 3.3.1 [P4], 4.1.1 [P2], 4.1.2 [P5], 5.1.2 [P2], 5.1.3 [P6], 5.2.1 [P3], 5.3.1 [P3], 5.4.1 [P4].

Create Performance Task

This section serves to fulfill the Performance Task requirements of the AP Computer Science Principles exam. The Create Performance Task will account for 24% 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. During this performance task, students will demonstrate their ability to work individually to design and develop a functional program for solving a problem and/or self-expression.

Create Performance Task Schedule

Topic	Tasks	Learning Objectives
Create – Applications from Ideas 12 hours of class time required	Identify Project Ideas	
	Develop, Implement, and Test Program	LO 1.2.1, LO 1.2.2, LO 1.2.3, LO 1.2.4, LO 2.2.1, LO 2.2.2, LO 4.1.2, LO 5.1.1, LO 5.1.2, LO 5.1.3, LO 5.3.1, LO 5.4.1, LO 5.5.1
	Create Video of Program	LO 1.2.1, LO 1.2.2, LO 1.2.3
	Write Responses on Program	LO 1.2.1, LO 1.2.2, LO 1.2.3, LO 5.2.1, LO 5.3.1
	Submit “Create” Task Program, Video, and Written Responses	

Create Performance Task Topics

Creative Development (P1, P2, P3, P4, P5, P6)

- Students will individually and/or collaboratively design, implement, and test a program designed to solve a problem of interest to them.
- Students will document the functionality of their program and reflect on its development process.

Create – Applications from Ideas Performance Task (P1, P2, P3, P4, P5, P6)

- This project will encompass 12 hours of in-class, independent and/or collaborative work.
- Each student will design, implement, and test a program that solves a problem of personal interest to the student.
- Each student will describe and reflect on their role in the development of the program.
- Students will make a one-minute video demonstrating the use and functionality of the program.
- Students may work collaboratively on their project, but each student will be solely responsible for developing at least one significant part of their program.
- The product of this project, including the program, video, and written responses, will serve as part of the student's formal submission to the College Board for the AP Computer Science Principles exam.

Unit 5: Big Data

One of the most powerful applications of computational thinking relates to the creation and analysis of large datasets. 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 techniques such as statistical analysis, data mining, clustering, classification, automated summarization, modeling and simulation 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 presentation.

The final activity described above is the one major project in this unit: the TEDxKinda Project.

Unit 5 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Data Science	Introduction to Big Data	1.2, 3.1, 3.2, 7.1, 7.2, 7.3	1.2.5, 3.1.1, 3.1.3, 3.2.1, 3.2.2, 7.1.1, 7.2.1, 7.3.1
	Usability and Usefulness of Data	3.1, 3.2	3.1.1, 3.2.2
Data Aggregation	Collection	3.2, 7.3	3.2.2, 7.3.1
	Extraction	3.1, 3.2, 7.5	3.1.1, 3.2.1, 7.5.1
	Data Storage and Persistence	3.2, 3.3, 7.3	3.2.1, 3.2.2, 3.3.1, 7.3.1
Big Picture	Wisdom of the Crowd	3.1, 3.2, 5.1, 7.1, 7.2	3.1.1, 3.1.2, 3.2.1, 3.2.2, 5.1.1, 7.1.1, 7.1.2, 7.2.1
	Data Breaches	3.2, 3.3, 7.3	3.2.2, 3.3.1, 7.3.1
Data Analysis	Statistical Analysis	3.1, 3.2, 7.1, 7.2	3.1.1, 3.1.3, 3.2.1, 7.1.1, 7.2.1
	Data Mining	3.1, 3.2, 7.2, 7.3	3.1.1, 3.2.1, 3.2.2, 7.2.1, 7.3.1
Models and Simulations	Models and Simulations	2.3	2.3.1, 2.3.2

Unit Project	TEDxKinda Project	1.2, 3.1, 3.2, 7.1, 7.2, 7.3, 7.5	1.2.1, 1.2.2, 1.2.4, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 7.1.1, 7.1.2, 7.2.1, 7.3.1, 7.5.1
Supplemental Data Analysis	Clustering	3.1, 3.2	3.1.1, 3.1.3, 3.2.1
	Anomaly Detection	3.1, 3.2, 7.1	3.1.1, 3.1.3, 3.2.1, 3.2.2, 7.1.1
	Regression	3.1, 3.2, 7.1	3.1.1, 3.1.3, 3.2.1, 3.2.2, 7.1.1
	Classification	3.1, 3.2	3.1.1, 3.1.3, 3.2.1, 3.2.2
	Automatic Summarization	3.1, 3.2	3.1.3, 3.2.1, 3.2.2

Unit 5 Topics

Data Science [EU 1.2, EU 3.1, EU 3.2, EU 7.1, EU 7.2, EU 7.3] [LO 1.2.5, LO 3.1.1, LO 3.1.3, LO 3.2.1, LO 3.2.2, LO 7.1.1, LO 7.2.1, LO 7.3.1] (P1, P3, P4, P5)

- Students will relate the impact of computing to ubiquitous and large-scale data processing.
- Students will explore the ways that patterns within large data sets can be used in a predictive manner.
- Students will discuss the risks and benefits of drawing conclusions from patterns found in large data sets.
- Students will combine visuals, content knowledge, and interaction to create a dynamic infographic that clearly communicates discrete information about a data set.
- Students will identify the characteristics that differentiate usable data from unusable data.
- Students will identify the characteristics that differentiate useful data from useless data.

Data Aggregation [EU 3.1, EU 3.2, EU 3.3, EU 7.3, EU 7.5] [LO 3.1.1, LO 3.2.1, LO 3.2.2, LO 3.3.1, LO 7.3.1, LO 7.5.1] (P1, P3, P4)

- Students will explore the purposes of various processing tasks, including collection, knowledge extraction, and data storage.
- Students will identify multiple techniques for data collection, both on and off of the Internet.
- Students will analyze the characteristics of structured and unstructured data.
- Students will extract structured information from unstructured data.
- Students will examine methods of extracting information from online sources, including structured and unstructured search engines, screen scrapers, and spiders.
- Students will explore the basic features and functionality of modern relational databases.
- Students will debate the implications of large-scale data storage and data persistence on privacy and utility, including the costs associated with each.

Big Picture [EU 3.1, EU 3.2, EU 3.3, EU 5.1, EU 7.1, EU 7.2, EU 7.3] [LO 3.1.1, LO 3.1.2, LO 3.2.1, LO 3.2.2, LO 3.3.1, LO 5.1.1, LO 7.1.1, LO 7.1.2, LO 7.2.1, LO 7.3.1] (P1, P2, P3, P4, P6)

- Students will apply the technique of crowdsourcing to a novel data collection problem.
- Students will examine the security risks and responsibilities assumed by companies that collect and store sensitive personal data.
- Students will examine the causes and impact of data breaches involving sensitive personal data.

Data Analysis (including Supplemental)[EU 3.1, EU 3.2, EU 7.1, EU 7.2, EU 7.3] [LO 3.1.1, LO 3.1.3, LO 3.2.1, LO 3.2.2, LO 7.1.1, LO 7.2.1, LO 7.3.1] (P1, P3, P4, P5)

- Students will analyze the tradeoff of utility and confidence in descriptive, predictive, and prescriptive data analysis.
- Students will investigate traditional statistical hypothesis testing and exploratory data analysis.
- Students will investigate the use of data mining in the discovery of patterns in large data sets.
- Students will examine the use of cluster analysis, anomaly detection, regression analysis, and data classification in the processing of large data sets.
- Students will use automatic summarization tools to create computer-generated summaries of a large data set.

Models and Simulations [EU 2.3] [LO 2.3.1, LO 2.3.2] (P3)

- Students will use models and simulations to represent phenomena.
- Students will explore how models may use different abstractions or levels of abstraction depending on the objects or phenomena being posed.
- Students will utilize models and simulations to formulate, refine, and test hypotheses.
- Students will examine how simulations mimic real world events without the cost or danger of building and testing the phenomena in the real world.

Unit Project [EU 1.2, EU 3.1, EU 3.2, EU 7.1, EU 7.2, EU 7.3, EU 7.5]
The TEDxKinda Project is a collaborative activity in this unit. In this data analysis project, students will work together to select and analyze a large data set, then develop a TED-style presentation to present the implications of that data. Students will:

- collaborate in groups to analyze public data sets and extract insightful information and new knowledge using a number of big data analysis techniques and tools.
- evaluate and justify the appropriateness of the chosen data set(s).
- construct informative and aesthetically pleasing data visualizations.
- write a script and prepare speaker notes for a formal presentation of the findings.
- cite all online and print sources used in the research and presentation preparation.
- deliver a TED-style presentation discussing the data analysis and findings using appropriate terminology.

This project encompasses the following College Board curricular requirements: LO 1.2.1 [P2], LO 1.2.2 [P2], LO 1.2.4 [P6], LO 3.1.1 [P4], LO 3.1.2 [P6], LO 3.1.3 [P5], LO 3.2.1 [P1], LO 3.2.2 [P3], LO 7.1.1 [P4], LO 7.1.2 [P4], LO 7.2.1 [P1], LO 7.3.1 [P4], LO 7.5.1 [P1].

Unit 6: Innovative 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 many of the key roles that technology plays in their lives, including social networking, online communication, search, commerce, and news, examining 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.

There is no major project in this unit, but there are several post-lesson opportunities for students to apply their learning.

Unit 6 Schedule

Topic	Lesson	Enduring Understandings	Learning Objectives
Big Picture	Defining a Computing Innovation	7.1, 7.3, 7.4	7.1.1, 7.3.1, 7.4.1
Implications of Computing	Global Impact	3.3, 6.1, 7.1, 7.4	3.3.1, 6.1.1, 7.1.1, 7.4.1
	Impact of Internet Access	3.2, 7.1, 7.2	3.2.1, 7.1.1, 7.2.1
	Cloud Computing	7.1, 7.3	7.1.1, 7.1.2, 7.3.1
Big Picture	The Digital Divide	7.4	7.4.1
The Internet	Internet in Action	6.1, 6.2	6.1.1, 6.2.1, 6.2.2
	Communication Protocols	5.3, 6.1, 6.2	5.3.1, 6.1.1, 6.2.2
	Internet Protocols	2.2, 6.1, 6.2, 6.3	2.2.3, 6.1.1, 6.2.1, 6.2.2, 6.3.1
Cryptography	Encryption	6.3, 7.3	6.3.1, 7.3.1
Big Picture	Net Neutrality	7.3, 7.4	7.3.1, 7.4.1
Cybersecurity	Cybersecurity	3.3, 6.3	3.3.1, 6.3.1

Interconnectedness in Computing	World Wide Web	2.2, 6.2	2.2.3, 6.2.2
	Distributed Computing	7.1	7.1.1, 7.1.2
	Internet of Things	7.1	7.1.1
	Ethics of Autonomous Technology	7.4	7.4.1

Unit 6 Topics

Big Picture [EU 7.1, EU 7.3, EU 7.4] [LO 7.1.1, LO 7.3.1, LO 7.4.1] (P1, P4)

- Students will examine computing innovations and consider their impact on the economy, society, culture and environment.
- Students will investigate the socioeconomic causes and effects related to the digital divide.
- Students will discuss the benefits and risks of open versus closed platforms.

Implications of Computing [EU 3.2, EU 3.3, EU 6.1, EU 7.1, EU 7.2, EU 7.3, EU 7.4] [LO 3.2.1, 3.3.1, 6.1.1, 7.1.1, 7.1.2, 7.2.1, 7.3.1, 7.4.1] (P1, P3, P4)

- Students will explore the ways that innovations in digital technology can impact the lives of individuals and communities.
- Students will analyze the role that digital technology plays in their everyday lives.
- Students will analyze the role that digital technology plays in their social communications and interactions.
- Students will explore the impact that instant access to global search, news, and information has had on individuals and communities.
- Students will analyze the benefits and risks of cloud computing.

The Internet [EU 2.2, EU 5.3, EU 6.1, EU 6.2, EU 6.3] [LO 2.2.3, LO 5.3.1, LO 6.1.1, LO 6.2.1, LO 6.2.2, LO 6.3.1] (P1, P3, P4, P5)

- Students will examine the overall design and architecture of the Internet.
- Students will explore the role of servers, routers, gateways, and clients.
- Students will examine the domain name system and its role in network routing.
- Students will examine a number of standard network protocols, including IP, TCP, UDP, SMTP, HTTP, and FTP.
- Students will investigate the series of components and events that are involved in the transmission of an email or SMS text over the network.
- Students will investigate the series of components and events that are involved in the transmission of an HTML request from a Web browser.

Cryptography [EU 6.3, EU 7.3] [LO 6.3.1, LO 7.3.1] (P1, P4)

- Students will identify the needs and applications of cryptography in our digital world.
- Students will encode and decode messages using common cryptographic techniques.
- Students will examine the mathematical foundation of cryptography.
- Students will analyze the differences between symmetric (single-key) encryption and asymmetric (public key) encryption.
- Students will examine the features of open and closed platforms and consider the role cryptography plays in systems security.

Cybersecurity [EU 3.3, EU 6.3] [LO 3.3.1, LO 6.3.1] (P1, P4)

- Students will examine a number of common threats to cybersecurity, including distributed denial of service attacks (DDoS), phishing, viruses, and social engineering.
- Students will identify the needs for robust cybersecurity.
- Students will analyze the software, hardware, and human components of cybersecurity.
- Students will analyze the function and effectiveness of common cybersecurity solutions, including antivirus software and firewalls.

Interconnectedness in Computing [EU 2.2, EU 6.2, EU 7.1, EU 7.4] [LO 2.2.3, LO 6.2.2, LO 7.1.1, LO 7.1.2, LO 7.4.1] (P1, P3, P4)

- Students will investigate the origins and applications of the World Wide Web.
- Students will analyze the impact of hyperlinked documents on how individuals find, acquire, and learn new information.
- Students will analyze the legal, social, and commercial impact that the World Wide Web has had on society.
- Students will examine the roles and applications of distributed computing.
- Students will investigate and extrapolate from recent advances in computing to make predictions about the capabilities of future technologies.
- Students will analyze how future technologies might impact individuals and societies.
- Students will examine the legal and ethical implications of autonomous technology.

Mini-Explore Task Module

This mini-performance task module is a multi-day activity that gives students a chance to deepen their understanding of the AP CSP Explore Task. They begin the module by exploring the requirements of the task itself. Then, they proceed to evaluate sample student submissions against the official College Board rubric. Then, students work through a small exercise sequence that helps them practice applying the project requirements. In this exercise sequence, students will:

- identify and describe a computing innovation
- assess the impact of the innovation
- outline how the innovation uses data
- identify a data security concern related to the innovation
- build a computational artifact to illustrate, represent or explain the innovation's intended purpose, function, or effect.
- review a peer's work and provide feedback, based on the official Create Task rubric.

This project encompasses the following College Board curricular requirements: [EU 1.2, EU 3.3, EU 5.2, EU 5.4, EU 7.1, EU 7.2, EU 7.3, EU 7.4] [LO 1.2.1 [P2], LO 1.2.2 [P2], LO 1.2.3 [P2], LO 1.2.5 [P4], LO 3.3.1 [P4], LO 5.2.1 [P3], LO 5.4.1 [P4], LO 7.1.1 [P4], LO 7.2.1 [P1], LO 7.3.1 [P4], LO 7.4.1 [P1], LO 7.5.1 [P1], LO 7.5.2 [P5]].

Explore Performance Task

This section serves to fulfill the Performance Task requirements of the AP Computer Science Principles exam. The Explore Performance Task will account for 16% 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. During the Explore – Impact of Computing Innovations Performance Task, students will demonstrate their ability to conduct independent research into an innovative technology and intelligently discuss its impact and influence on society as a whole.

Explore Performance Task Schedule

Topic	Tasks	Learning Objectives
Explore – Impact of Computing Innovations 8 hours of class time required	Identify Topic of Research	
	Conduct Independent Research	LO 1.2.5, LO 3.3.1, LO 7.1.1, LO 7.2.1, LO 7.3.1, LO 7.4.1, LO 7.5.2
	Write Responses to Prompts	LO 5.2.1, LO 5.4.1
	Create Computational Artifact	LO 1.2.1, LO 1.2.2, LO 1.2.3, LO 7.1.1, LO 7.2.1, LO 7.3.1, LO 7.4.1
	Write Support for Computational Artifact	LO 1.2.2, LO 1.2.3, LO 1.2.5, LO 7.1.1, LO 7.2.1, LO 7.3.1, LO 7.4.1, LO 7.5.1
	Submit "Explore" Task Computational Artifact and Written Responses	

Explore Performance Task Topics

Exploration and Research (P1, P2, P3, P4, P5)

- Students will conduct independent research into a technological innovation of their choice.
- Students will examine the social, economic, and cultural impact of their chosen technological innovation.
- Students will examine how their chosen technological innovation consumes, produces, and/or transforms data.
- Students will identify and discuss concerns about data storage, data privacy, or data security with regard to their chosen technological innovation.

Explore – Impact of Computing Innovations Performance Task (P1, P2, P4, P5)

- This project will encompass eight hours of in-class independent research and work.
- Each student will investigate a computing innovation of his/her choice that has had a significant impact on society, economy, or culture.

- Each student will produce a computational artifact that describes the intended purpose and function of the computing innovation and demonstrates how it fulfills that purpose.
- Each student will document the development process, tools, and techniques used in creating the computational artifact.
- Each student will identify and explain the beneficial and harmful effects of the computing innovation on society, economy, or culture.
- Each student will identify and discuss how the computing innovation consumes, produces, and/or transforms data and address concerns about data storage, data privacy, or data security with regard to their chosen technological innovation.
- Each student will thoroughly cite the sources used in conducting their research of the computing innovation.
- The product of this project, including the computational artifact and written responses, will serve as part of the student's formal submission to the College Board for the AP Computer Science Principles exam.

Unit 7: The AP Exam

This final unit provides preparation resources for all components of the AP exam. Students review the design of the assessment and work through a practice multiple choice sequence taken from the College Board’s AP Computer Science Principles Course & Exam Description document. Students also have access to resources that support the completion and submission of the Performance Task requirements. As students practice and prepare for the exam with this module and other past curricular content, they should consider each of the seven big ideas and six computational thinking practices central to this course.