CollegeBoard AP®

SYLLABUS DEVELOPMENT GUIDE

AP® Computer Science Principles

The guide contains the following sections and information:

Curricular Requirements

The curricular requirements are the core elements of the course. A syllabus must provide explicit evidence of each requirement based on the required evidence statement(s).

Required Evidence

These statements describe the type of evidence and level of detail required in the syllabus to demonstrate how the curricular requirement is met in the course.

Note: Curricular requirements may have more than one required evidence statement. Each statement must be addressed to fulfill the requirement.

Clarifying Term(s)

Highlight and define terms in the syllabus development guide that may have multiple meanings.

Samples of Evidence

For each curricular requirement, three separate samples of evidence are provided. These samples provide either verbatim evidence or clear descriptions of what acceptable evidence could look like in a syllabus.

Curricular Requirements

CR1	The teacher and students have access to college-level computer science resources, in print or electronic format.	*See page:* 3
CR2	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 (CED).	*See page:* 4
CR3	The course provides opportunities to develop student understanding of the big ideas, as outlined in the AP Course and Exam Description.	*See page:* 7
CR4	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 1: Computational Solution Design.	*See page:* 10
CR5	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 2: Algorithms and Program Development.	*See page:* 11
CR6	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 3: Abstraction in Program Development.	*See page:* 12
CR7	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 4: Code Analysis.	*See page:* 13
CR8	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 5: Computing Innovations.	*See page:* 14
CR9	The course provides opportunities for students to develop the skills related to Computational Thinking Practice 6: Responsible Computing.	*See page:* 15
CR10	The course provides a minimum of three opportunities for students to investigate different computing innovations.	*See page:* 16
CR11	Students are provided at least 12 hours of dedicated class time to complete the AP Create Performance Task.	*See page:* 18

Curricular Requirement 1

The teacher and students have access to college-level computer science resources, in print or electronic format.

Required Evidence

- ☐ The syllabus must list a college-level resource, which may include a computer science textbook, website, article, or video.

Samples of Evidence

1. The syllabus cites a textbook from the AP® Example Textbook List for computer science principles.

2. The syllabus cites Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits: Your Life, Liberty, and Pursuit of Happiness After the Digital Explosion.* Addison-Wesley. **www.bitsbook.com/thebook**

3. The syllabus lists the college-level website ACM TechNews and Runestone Academy's "How to think like a computer scientist: Interactive edition."

Curricular Requirement 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.

Required Evidence

- ☐ The syllabus must include an outline of course content by unit or module using any organizational approach with the associated big idea(s) to demonstrate the inclusion of required course content. The outline must include labeling of all five big ideas.

Note: Unit/module titles alone are insufficient evidence.

Samples of Evidence

1. Unit 1: The Internet.

   Students will learn how computers represent data, convert between binary numbers and decimal numbers, and compress data. They will learn how people participate in problem-solving, how to make systems fault-tolerant, and how information is sent over networks like the internet through computer network connectivity. **(DAT, IOC, CSN)**

   Unit 2: Language of Computing.

   Students will be introduced to the incremental and iterative software development process, variables and expressions, and algorithms for selection and iteration. Students will learn how to interpret procedure calls and generate random numbers for simulations. **(CRD, AAP)**

   Unit 3: Building Blocks of Algorithms.

   Students learn how collaboration improves innovation, how to identify program input and output, and how to implement boolean expressions. They will also learn about selection and iteration and begin thinking about how innovations have had both beneficial and harmful impacts. **(CRD, AAP, IOC)**

   Unit 4: Solving Problems with Computing.

   Students will learn how innovations are developed by groups of people, and understand how to identify and correct errors in programs, combine code or modify algorithms to create new innovations, develop procedural abstractions through writing procedures, further explore how innovations can have bias and impact beyond their original intent, and grasp parallel and distributed computing solutions. **(CRD, AAP, IOC, CSN)**

   Unit 5: Abstractions in Programming.

   Students will learn about effective skills for collaboration, the existence of undecidable problems, what information can be extracted from data and metadata, and the privacy risks for collecting and storing personal data. They will also learn the skills to develop data abstractions through the use of lists and strings, utilize existing code and libraries, and identify test cases and expected outcomes for programs. **(CRD, AAP, DAT, IOC)**

   Unit 6: Real-World Simulations.

   Students will learn how to describe what a program does and acknowledge the contributions of others, determine if an algorithm runs in reasonable time or not, extract information from data, and protect computing resources from being misused. They will also learn about searching algorithms, the legal issues of computing, and how computers can represent real-world phenomena. **(CRD, AAP, DAT, IOC)**

2. Unit 1: Impact of Computing Innovations

   Students will learn the course big ideas, effective collaboration techniques, and different types of communication while exploring innovations **(IOC, DAT, CSN)**, diagraming and developing algorithms **(AAP)**, and engaging in programming **(CRD, DAT, AAP, IOC)**

   Unit 2: Programming Foundations

   Students will learn the software development process while creating programs via guided instruction and/or pair programming **(CRD, AAP)**

   Unit 3: Algorithms in Programming

   Students will learn how algorithms are implemented in programming and how abstraction manages complexity in programs **(CRD, AAP)**

   Unit 4: Exploring Innovations I

   Students will learn to analyze a computing innovation's purpose and function as well as how an artifact can represent an innovation **(CRD, DAT, IOC)**

   Unit 5: Abstraction in Programming I: Procedural

   Students will learn to develop procedural abstraction(s) to manage complexity in programs and to prepare program documentation **(CRD, AAP)**

   Unit 6: Exploring Innovations II

   Students will learn to collaboratively investigate computing innovations for beneficial and harmful effects using a compare/contrast approach **(CRD, IOC)**

   Unit 7: Abstraction in Programming II: Data

   Students will learn to utilize data abstraction to provide user engagement and manage complexity in programs **(CRD, DAT, AAP)**

   Unit 8: Exploring Innovations III

   Students will learn to analyze a computing innovation based on its data utilization with consideration regarding concerns related to data while preparing a mind map artifact of hypotheses and findings (**CRD, DAT, IOC**, Explore CR)

   Unit 9: Program Development: Collaboration and Test Cases

   Students will learn to collaboratively develop an interactive program that includes each type of algorithm implementation (sequencing, selection, and iteration) and both types of abstraction (procedural and data) **(CRD, DAT, AAP)**

   Unit 10: Computing Systems and Networks

   Students will learn to identify and use internet systems including network protocols, layers of abstraction on the internet, and data transfer **(DAT, CSN, IOC)**

   Unit 11: Program Development: Simulations and Models

   Students will learn to create a program that simulates an everyday task and demonstrates functionality while preparing deliverables of the AP Create Performance Task (Create PT) for teacher and peer feedback **(CRD, DAT, AAP, IOC)**

3. Course Readings:

   Abelson, Hal, Ken Ledeen, and Harry Lewis. Addison-Wesley. *Blown to Bits: Your Life, Liberty, and Pursuit of Happiness After the Digital Explosion.*

   Chapter 1: Digital Explosion (DAT, IOC)

   - Bits are Everywhere

Chapter 2: Naked in the Sunlight (Privacy Lost, Privacy Abandoned) (DAT, IOC, CSN)

- Privacy Lost or Abandoned?
- Footprints and Fingerprints
- Technology Change and Lifestyle Change

Chapter 3: Ghosts in the Machine (DAT, IOC)

- Secrets and Surprises of Electronic Documents
- Hiding Information in Images

Chapter 4: Needles in the Haystack (CSN, DAT, IOC)

- Google and Other Brokers in the Bits Bazaar
- Who Pays for What?
- Tracking Searches

Chapter 5: Secret Bits (DAT, IOC)

- How Code becomes Unbreakable
- Encryption

Appendix: The Internet as System and Spirit (CSN)

- The Internet as a Communication System
- How the Internet Works

Computer Science Illuminated, 6th Edition 2016, Authors: Nell Dale (University of Texas, Austin); John Lewis (Virginia Tech, Blacksburg).

Chapter 1: Laying the Groundwork (IOC)

- Describe the layers of a computer system
- Describe the concept of abstraction and its relationship to computing
- Describe the history of computer hardware and software
- Describe the changing role of the computer user
- Distinguish between systems programmers and applications programmers
- Distinguish between computing as a tool and computing as a discipline

Chapter 2: The Information Layer (DAT)

- Distinguish among categories of numbers
- Describe positional notation
- Convert numbers in other bases to base 10
- Convert base-10 numbers to numbers in other bases
- Describe the relationship between bases 2, 8, and 16
- Explain the importance to computing of bases that are powers of 2

Chapter 7: Problem Solving and Algorithms (AAP, CRD)

- Describe the computer problem-solving process
- Distinguish between types
- Describe data-structuring mechanisms
- Distinguish between an unsorted array and a sorted array
- Distinguish between a selection sort and an insertion sort

Chapter 15: Networks (CSN)

- Describe the core issues related to computer networks
- List various types of networks and their characteristics
- Explain various topologies of local-area networks
- Explain why network technologies are best implemented as open systems
- Compare and contrast various technologies for home internet connections

Curricular Requirement 3

The course provides opportunities to develop student understanding of the big ideas.

Required Evidence

- ☐ The syllabus must include at least five activities, each of which is explicitly related to one or more of the five big ideas. Each big idea must be included in at least one activity. Each activity must be labeled with the related big idea(s).

Note: The Create Performance Task is a summative assessment and cannot be included as one of these five student activities.

Samples of Evidence

1. **Creative Development (CRD)**

   Throughout the course, students are required to submit four programming assignments. Each assignment requires submission of a written explanation of the program design process including problems encountered during program development and implementation, documentation of any help/assistance needed in the development process, and the program code that successfully accomplishes the tasks described in the assignment specifications. Sample programs assigned:

   a. Simple two-player game
   b. Simple simulation of real-world phenomena (elevator)
   c. Random number simulation (lottery and respective payouts)
   d. Calculation of final grades

   Data (DAT)

   Class discussions of data representation demonstrate the comparisons of memory (bits) required to store text, images, and videos. Students will complete activities involving simple data compression using multiple compression algorithms.

   Algorithms and Programming (AAP)

   Students work in pairs to describe two different algorithms to solve the following problem: Fill a 20-element array/array list with random integers from 1 to 100 (inclusive) ensuring that there are no duplicate values in the array. Algorithms are shared in class. Each team then implements their algorithm of choice in a well-documented program.

   Computing Systems and Networks (CSN)

   Routing and deadlock in networks is introduced with Computer Science Unplugged—The Orange Game. Class is divided into teams of 5–6 students. This is a co-operative problem-solving game. The aim is for each person to end up holding the oranges labeled with their own letter.

   a. The students are labeled with a letter of the alphabet. There are two oranges with each student's letter on them, except for one student, who only has one corresponding orange to ensure that there is always an empty hand.
   b. Labeled oranges are distributed.
   c. The students pass the oranges around until each has the oranges labeled with their letter of the alphabet. Two rules: a) Only one orange may be held in a hand. b) An orange can only be passed to an empty hand of an immediate neighbor.

Impact of Computing (IOC)

Students will work in pairs to investigate a computing innovation of their choice. They will research the innovation's functionality and beneficial and harmful effects. The students will then present their findings to the class in a 5–10 minute presentation.

2. **Big Idea 1: Creative Development**

   Students will create a quiz game project that features an engaging user interface and tests the user on a subject or topic of the programmer's choice, pulling questions and answers from lists. **(CRD)**

   Big Idea 2: Data

   Students will create an encoded message to a classmate converting from English alphabet to bits using the ASCII table and decimal number to binary conversion; recipients will decode messages and create a reply using a Caesar cipher shift encryption providing an algorithm to hint at the shift. **(DAT)**

   Big Idea 3: Algorithms and Programming

   Students will create a random number generator app in a both block- and text-based language. The programs will utilize algorithms to provide reliable output and will include a procedure to generate the random number. Students will compare and contrast programming in a block-based and text-based language. **(AAP)**

   Big Idea 4: Computing Systems and Networks

   Students will work in collaborative teams of 2–3 to simulate a packet ping in the school. Teams will be given an "encrypted" ASCII/binary/hexadecimal room number destination "IP address" to ping, and then will travel through the school to find the location where their fetch packet will be traded for a return packet, which contains another "encrypted" destination "IP address" to the original room. Upon return, packets are ordered as returned but not opened until all are returned. Once opened, the packets contain a card with an ASCII number that once converted yields a letter which when placed in proper sequences forms a word related to the Computing Systems and Networks big idea. **(CSN)**

   Big Idea 5: Impact of Computing

   Students will work in collaborative pairs to use only social media in order to find out as much as possible about the last seven days in the life of a well-known person. Teams will create a timeline artifact with dates, times, locations from the publicly shared data. Teams will prepare a reflection discussing the impact of computing on security and privacy, including the beneficial and harmful uses and impact of public data. **(IOC)**

3. Unit 1: The Internet—Students will simulate how information is sent and routed over the internet by passing index cards to represent packets around the room. Each student acts like a router and passes the information to other routers to get to the final destination. **(CSN)**

   Unit 2: Language of Computing—In groups, students will play board games for 10 minutes, taking note of the steps, decisions, and iterations that occur. Then they will create an algorithm for how to play the game in the form of a diagram or pseudocode. They will then ask another group to play the game, strictly following their steps to identify any steps that were missing. A debrief will be conducted at the conclusion of the lesson to ensure all students understand the importance of including all the steps in the correct order for a program as well as to recognize the selection and iteration that is naturally occurring. **(AAP)**

Students will complete Activity 1: Selecting Computing Innovations in the *AP Computer Science Principles Explore Curricular Requirements Teacher Resources.* **(CRD)**

Unit 3: Building Blocks of Algorithms—Students will complete Activity 2: Analyzing Data and Computing Innovations in the *AP Computer Science Principles Explore Curricular Requirements Teacher Resources.* **(AAP, CRD, IOC)**

Unit 4: Solving Problems with Computing—Students will form a group of 4–5. They will identify a problem that affects them and then in pairs work to design a program or part of a program as a solution. Students will then regroup as their original group of 4–5 to compare their solution designs to see if they would yield equivalent results. In this larger group, they will determine the best design or combine elements of each design for a final design that will be implemented. Students will be required to incorporate procedural abstraction into their program. To accompany their program submission, students will answer a series of prompts similar to those that will be required for the Create PT. **(CRD, AAP)**

Students will complete Activity 3: Analyzing Impact of Computing in the *AP Computer Science Principles Explore Curricular Requirements Teacher Resources.* **(IOC)**

Unit 5: Abstractions in Programming—Students will work in small groups to develop a question they are interested in learning the answer to. Using survey data or publicly available data sets, students will analyze this data or metadata to determine the answer to their question. **(DAT)**

Students will write a program that uses a list to store and analyze data. To accompany their program submission, students will answer a series of prompts similar to those that will be required for the Create PT. **(AAP)**

Unit 6: Real-World Simulations—Students will be asked to develop a simulation of something in the real world. As part of the design of this program, students will need to develop a list of test cases that will be used to test the simulation to see if it is functioning properly. This will allow students to make adjustments to the simulation to make it more accurate. **(CRD, AAP)**

Curricular Requirement 4

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 1: Computational Solution Design, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students design or evaluate computational solutions for a purpose. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Samples of Evidence

1. Unit 4: Solving Problems with Computing—Students will form a group of 4–5. They will identify a problem that affects them and then in pairs work to design a program or part of a program as a solution. Students will then regroup as their original group of 4–5 to compare their solution designs to see if they would yield equivalent results. In this larger group, they will determine the best design or combine elements of each design for a final design that will be implemented. **(Computational Thinking Practice 1)**

2. Students are given a variety of board games from which they choose one. For the chosen board game, students design a program that will allow two players to play a simplified version of the game on a computer. **(P1)**

3. Unit 7: Students will create a chatbot program that uses a list or lists to hold random phrases that are generated based on user responses to questions in the program. **(CTP 1)**

Curricular Requirement 5

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 2: Algorithms and Program Development, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students develop **and** implement algorithms. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Samples of Evidence

1. Unit 2: Language of Computing—In groups, students will play board games for 10 minutes, taking note of the steps, decisions, and iterations that occur. Then they will create an algorithm for how to play the game in the form of a diagram or pseudocode. They will then ask another group to play the game, strictly following their steps to identify any steps that were missing. A debrief will be conducted at the conclusion of the lesson to ensure all students understand the importance of including all the steps in the correct order for a program as well as to recognize the selection and iteration that is naturally occurring. **(Computational Thinking Practice 2)**

2. Students work in pairs to design two different algorithms to choose 20 integers between 1 and 100 without duplicates. They will then compare the algorithms regarding the number of steps taken to solve the problem and the amount of extra space (additional memory locations) needed in each solution. Finally, they will choose one of the two algorithms to implement and test in Python. **(P2)**

3. Unit 2: Students will diagram a program to yield a random number based on the user's input of a desired minimum (MIN) and maximum (MAX) including the ability to generate another number or exit once a result is obtained. Based on the diagram, students will develop algorithms to receive the user MIN and MAX then sequentially verify those inputs before executing a function that receives the inputs as parameters. Finally, the program will display the random number followed by a prompt to generate another or exit. Students will integrate various sequencing, selection, and/or iteration algorithms to create this program. **(CTP 2)**

Curricular Requirement 6

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 3: Abstraction in Program Development, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students develop programs that incorporate abstractions. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Samples of Evidence

1. Unit 5: Abstracting in Programming—Students will write a program that uses a list to store and analyze data. To accompany their program submission, students will answer a series of prompts similar to those that will be required for the Create PT. **(Computational Thinking Practice 3)**

2. Students will play the Lightbot game on the computer. After successfully completing several levels, students discover that they need to create a procedure (abstraction) to succeed (they have only room for 12 statement invocations in the game). They must simplify their code by placing repeated code in a procedure and calling that procedure multiple times. **(P3)**

3. Unit 5: Students will create a five-function calculator app that receives user input of two whole numbers and a desired function to produce a result. The program features procedural abstractions of these five functions: computeSum, computeProd, computeQuo, computeDiff, and computePower. Based on the user's selection, a procedure is called to act upon the two variables received as user input. **(CTP 3)**

Curricular Requirement 7

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 4: Code Analysis, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students evaluate **and** test programs or algorithms. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Clarifying Terms

Evaluate: may include explanations of code segments/programs, comparing and contrasting code segments, or determining results of code segments

Test: uses test cases or defined inputs to ensure an algorithm or program is producing expected outcomes

Samples of Evidence

1. Unit 6: Real-World Simulations Students will be asked to develop a simulation of something in the real world. As part of the design of this program, students will need to develop a list of **test cases** that will be used to test the simulation to see if it is functioning properly. This will allow students to make adjustments to the simulation to make it more accurate. During a gallery walk, students will **explain how their program functions** to visiting peers. **(Computational Thinking Practice 4)**

2. Students work in pairs to design two different algorithms to choose 20 integers between 1 and 100 without duplicates. They will then **compare** the algorithms regarding the number of steps taken to solve the problem and the amount of extra space (additional memory locations) needed in each solution. Finally, they will choose one of the two algorithms to implement **and test** in Python. **(P4)**

3. Unit 11: Students will work collaboratively to simulate an everyday task in order to model the impact of computing. Teams will demonstrate functionality on a **series of test cases**, give and receive feedback from peers, and present programs to peers while **comparing and contrasting** benefits of their programmed simulation when compared to live testing. **(CTP 4)**

Curricular Requirement 8

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 5: Computing Innovations, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students investigate computing innovations. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Samples of Evidence

1. Unit 4: Solving Problems with Computing - Students will complete Analyzing Impact of Computing in College Board's *AP Computer Science Principles Explore Curricular Requirements Teacher Resources.* **(Computational Thinking Practice 5)**

2. Students will research social media sites in the news and participate in an online forum (created for the class) by identifying a news article that focuses on the social media site and possible ethical/legal impacts on the individual or on society. **(P5)**

3. Unit 1: Students will identify a computing innovation from an assigned broad category (outdoors, pets, health, entertainment, etc.) Using their selected innovation, students will prepare a single slide artifact and present a 2–3 minute flash talk about the purpose, function, and data of the innovation. **(CTP 5)**

Curricular Requirement 9

The course provides opportunities for students to develop the skills related to Computational Thinking Practice 6: Responsible Computing, as outlined in the AP Course and Exam Description.

Required Evidence

- ☐ The syllabus must include a description of an activity or series of activities in which students contribute to an inclusive, safe, collaborative, or ethical computing culture. Each activity must be labeled with the related practice.

Note: The Create Performance Task is a summative assessment and cannot be included as one of these activities.

Clarifying Term

Safe computing: measures taken to keep private information secure

Samples of Evidence

1. Unit 2: Language of Computing – In groups, students will play board games for 10 minutes, taking note of the steps, decisions, and iterations that occur. Then they will create an algorithm for how to play the game in the form of a diagram or pseudocode. They will then ask another group to play the game, strictly following their steps to identify any steps that were missing. A debrief will be conducted at the conclusion of the lesson to ensure all students understand the importance of including all the steps in the correct order for a program as well as to recognize the selection and iteration that is naturally occurring. **(Computational Thinking Practice 6)**

2. Students will work in pairs to design a program or app (or enhance one that already exists) that has the potential to benefit society. The pair will then collaborative to create a simple version of the program (app) and include any necessary acknowledgements of borrowed code (APIs). The students will present a demo to the class along with a short presentation of how this app would benefit society. **(P6)**

3. Unit 10: Students will work collaboratively with a classmate to investigate an assigned topic related to data storage, data security, data privacy, and/or data transfer including tips and dangers. Each team will create one slide to be shared to an open cloud drive repository. This activity models crowdsourcing and provides the basis for summative whole-group discussion of the cumulative results of each team's work along with the benefits and risks of the open drive repository and related ethical practices. **(CTP 6)**

Curricular Requirement 10

The course provides a minimum of three opportunities for students to investigate different computing innovations.

Required Evidence

- ☐ The syllabus must explicitly identify and describe a minimum of three activities addressing different computing innovations. Within these three activities, students are required to address the following prompts at least once:
  - Explain beneficial and harmful effects of at least one computing innovation on society, economy, or culture.
  - Identify the data used in at least one computing innovation and explain how the data is consumed, produced, or transformed by the given computing innovation.
  - Identify data privacy, security, or storage concerns for at least one computing innovation.
- ☐ For each activity, use the label Computing Innovation (or CI) 1, 2, or 3 to identify the activity, and use the label A, B, or C to identify the prompt(s).
- ☐ If the syllabus uses all three activities from College Board's *AP Computer Science Principles Explore Curricular Requirements Teacher Resources*, the requirement is satisfied.

Samples of Evidence

1. Students complete the three activities outlined in College Board's *AP Computer Science Principles Explore Curricular Requirements Teacher Resources*:
   - Selecting Computing Innovations
   - Analyzing Data for Computing Innovations
   - Analyzing Impact of Computing

2. **Computing Innovation 1, Prompt A:** Students will participate in the Debate Carousel Activity where the prompt is: "Innovation (chosen by teacher) has impacted society in a beneficial and harmful way." Debate Team Carousel involves working in teams of four. Each student is given a paper divided into four boxes. Boxes are labeled: 1) make a claim and explain your rationale, 2) add supporting argument, 3) make a counter claim or provide a challenge to the claim, 4) read all blocks and write your opinion.

   A designated amount of time is given to the students to complete Box 1. The papers are then rotated. Students read what is written on their current paper and complete Box 2. Papers are rotated again. Students read what is written on their current paper and complete Box 3. Papers are rotated again. Students complete Box 4. A three minute discussion takes place within the teams. Papers are collected.

   Computing Innovation 2, Prompt B: Students work in teams of four. Each team chooses a computing innovation that all team members are familiar with. The students then use online resources to research the functionality of the innovation and how the innovation uses, consumes, or produces data. A designated amount of time is given for the students to work individually. Teams then use an "all write round robin" activity to share their findings. During a round robin, each student takes an equal amount of time to respond and all students record each response on their paper. After two rounds, a designated amount of time is given for the teams to discuss their findings. Each team shares with the class.

   Computing Innovation 3, Prompt C: Students are given worksheets on which are listed data, privacy, and security concerns for various computing innovations. Students are to identify the listed concern as a) a privacy concern, b) a security concern, or c) a storage concern.

3. Students will complete three innovation impact reports as follows:

   Students will work individually to explore a chosen innovation and related publications to determine its (1) purpose, (2) function, and (3) beneficial and (4) harmful impacts, and the (5) data input/transformed/output while creating an artifact that identifies each of the five areas of exploration. **(CI 1, Prompts A and B)**

   Students will work collaboratively in pairs to explore two types of the same innovation considering for each type the purpose, function, user interface, data input/transformed/output, and ratings and costs. From their analysis, teams will create an artifact that compares and contrasts functionality, beneficial effects, harmful effects, and overall usability of each while yielding a recommendation based on review of references and the team's research. **(CI 2, Prompts A and B)**

   Students will work individually to explore an assigned innovation by reviewing its purpose, function, effects, data, and user interface in order to identify potential security, storage, and privacy concerns related to the ownership or use of the innovation while creating a mind map of hypotheses and findings. **(CI 3, Prompts A, B, and C)**

Curricular Requirement 11

Students are provided at least 12 hours of dedicated class time to complete the AP Create Performance Task.

Required Evidence

- ☐ The syllabus must include an explicit statement that students are provided with at least 12 hours of class time to complete the Create Performance Task.

Clarifying Term

Hours of class time: at least 720 minutes provided in the classroom

Samples of Evidence

1. Twelve hours of class time will be provided for the Create Performance Task.
2. After completing Unit 10, students complete the Create Performance Task (12 hours in class).
3. Fifteen class periods (which is at least 12 hours of class time) will be provided for students to complete the Create Performance Task.