SAMPLE SYLLABUS #2

AP® Computer Science Principles

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.	*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 (CED).	*See pages:* 5, 6, 7, 9
CR4	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.	*See page:* 8
CR5	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.	*See page:* 8
CR6	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.	*See page:* 7
CR7	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.	*See page:* 7
CR8	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.	*See page:* 9
CR9	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.	*See page:* 10
CR10	The course provides a minimum of three opportunities for students to investigate different computing innovations.	*See pages:* 9, 10
CR11	Students are provided at least 9 hours of dedicated class time to complete the AP Create Performance Task.	*See page:* 5

Advanced Placement Computer Science Principles Sample Syllabus #2

Student Expectations: Students enter AP® Computer Science Principles (AP CSP) with a wide variety of backgrounds. Some students have had no formal computer science training. Others have taken an introduction to HTML and CSS or an introduction to BASIC programming in one of the introductory courses offered at the school. Some students have experienced summer programs that included introductory programming using one of several possible programming languages. We also have a small number of students who have completed AP Computer Science A (AP CSA) before entering AP Computer Science Principles or are concurrently enrolled in AP CSA. It is expected that all students entering AP Computer Science Principles will be knowledgeable about the topics taught in Algebra I, including functions, variables, and expressions.

Course Overview:

AP Computer Science Principles is designed to encourage a diverse group of students to explore computer science. Rather than limiting this introductory study to a single big idea—algorithms and programming—this course introduces students to a broad set of big ideas. These big ideas, which include algorithms and programming, are creative development, data, computing systems and networks, and the impact of computing. In addition, this course emphasizes the use of computational thinking practices for effective learning experiences and problem solving. These practices include computational solution design, algorithms and program development, abstraction in program development, code analysis, computing innovations, and responsible computing.

At our school, AP CSP is explored using an integrated learning approach. Students build their knowledge and understanding through participation in a wide variety of activities and explorations. These experiences are not broken into contiguous blocks of common content. Instead, they are integrated throughout the school year. Before, during, and after explorations, connections are made to the five big ideas at the core of the course. Activities encourage students to regularly apply the six computational thinking practices to their work. It is impractical to expect that students will fully understand any significant computer science topic after their first exposure to the subject matter. Students will reflect upon and expand their understanding as they revisit related topics throughout the course. Assessments will build on prior knowledge, and student performance expectations for specific topics will increase with each assessment.

The primary programming language used in this course is JavaScript. However, students are exposed to other programming options as well.

Throughout this course, students are guided toward personal discoveries and introduced to computer science topics that are related to current events and their own experiences. Often, students are drawn to these topics through the use of storytelling. The shared stories originate from community experiences and from current and former student experiences. They focus on topics related to popular culture, historical events, and other areas of student interest. Students are encouraged to pursue personal interests related to the presented materials, which often lead explorations in unanticipated directions. Student-initiated explorations are among the most valuable for both students and the instructor. Explorations for AP CSP are designed to spark interest, curiosity, enthusiasm, and enjoyment. After each exploration, discussions are organized to facilitate student reflections about what they have learned, what they have yet to learn, and what they wish to further explore in future studies. Both students and the instructor influence the final activities and related requirements associated with final assessments.

Big Ideas

BI1: Creative Development (CRD)

BI2: Data (DAT)

BI3: Algorithms and Programming (AAP)

BI4: Computing Systems and Networks (CSN)

BI5: Impact of Computing (IOC)

Computational Thinking Practices

CTP1: Computational Solution Design (Design and Evaluate)

CTP2: Algorithms and Program Development (Develop)

CTP3: Abstraction in Program Development (Develop)

CTP4: Code Analysis (Analyze)

CTP5: Computing Innovations (Investigate)

CTP6: Responsible Computing (Contribute)

Text and Supplemental Materials: CR1

1. Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits, Your Life, Liberty, and Pursuit of Happiness After the Digital Explosion.* Addison-Wesley
2. Nick Parlante, Stanford University, IntroComputing: A free website of materials based on Nick Parlante's Stanford CS101 course
3. ACM TechNews, computer science and related science, society, and technology news
4. Various websites that provide relatively short videos on topics relevant to computer science.

CR1
The syllabus must list a college-level resource, which may include a computer science textbook, website, article, or video.

Teaching Strategies:

In order to promote equity and access for all students, cooperative learning structures (CLS) are used regularly throughout the course to encourage 100% active participation in both speaking and listening activities. Students often work with a "shoulder partner," a table group of three or four students, or a larger group. Kinesthetic learning opportunities are provided throughout the course in order to address varied learning styles. The Socratic method (SM) is used regularly to allow students to share their insights, expand their understanding, and explore new related ideas in the context of their current work. Weekly reading assignments (WR) from the course resources are used to provide students with initial insights that are refined through probing questioning and discussion. Role play (RP) is used throughout the course to allow students to visualize and act out important algorithms. Computer animations (CA) are also used from various websites to reinforce student understanding. Early in the course, relatively short programming assignments are given on a weekly basis to allow students to apply their knowledge to code creation working individually (WI) and in small groups (SG). As the course progresses, students are required to work on group programming projects (GP) that include design exercises and code sharing.

Course Overview CR2

Exploration Theme	Hours of Exploration	Exploration Theme Summary
0: Introduction to Computer Science Principles	15	Computer science is introduced as a study of the five big ideas facilitated through the application of the six computational thinking practices. Various websites are used to inspire discussions and document creation related to current topics. Students read and discuss articles, respond in journal entries, and create artifacts that summarize their interests and knowledge. (CRD, IOC)
1: Creativity and Computing	20	Students write program code, primarily in JavaScript, throughout the school year. Much of this programming is focused on creating artifacts of interest to individual students. Manipulations range from creating patterns in images using mathematical computations, to algorithmic computations associated with image filtering. Students write programs in order to merge images using pixel selection algorithms and to combine multiple techniques into new, higher-level techniques. Students analyze and visualize data sets using computational techniques, explore JavaScript APIs, and create a dynamic website that facilitates the exploration of topics of personal interest. (CRD, DAT, AAP)
2: Identifying, Creating, and Using Abstractions for Enhanced Creativity and Problem Solving	10	Students are introduced to abstractions used to efficiently create effective programs. Everything that a student says or does involves the use of abstractions. It is important to help students become aware of abstractions and how they impact their ability to understand and navigate the world in which they live. In particular, AP CSP involves abstractions whose application enhances computational capabilities. (CRD, DAT, AAP, IOC)
3: Using Data to Gather Information and Create New Knowledge	10	Programming is used to help students effectively process and summarize data. Publicly available data sources are used for exploration, the discovery of information, and the creation of new knowledge. Students identify topics that interest them and search the web for data associated with those topics. Early in the course, students are provided data sources that are stored on the server used for their programming. This allows students to more easily process data in an effort to discover new information and knowledge. Later in the course, data sources must be entered into the programming environment in an acceptable format. This often requires computational manipulation of the data using text editors, spreadsheets, JavaScript programs, or statistics software packages. (CRD, DAT, AAP, IOC)
4: Exploring Algorithms for Enhanced Creativity and Problem Solving	10	Students present, analyze, and implement algorithms that are designed to accomplish specific tasks related to solving problems of personal interest and motivational artifact creation. While exploring image manipulations, students learn how to create gray scale images, produce enhanced color schemes based on original image color schemes, merge images, and transform images using reflections, rotations, and dilations. Students also generate row, column, diagonal, and checkerboard patterns, as well as mathematical patterns created using common functions from mathematics courses students have previously studied. They use their knowledge and experience to create new algorithms that accomplish their image manipulation goals. Students also apply their knowledge of algorithms in order to process data and gain new information and knowledge. (CRD, AAP)

CR2

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.

Unit/module titles alone are insufficient evidence.

Exploration Theme	Hours of Exploration	Exploration Theme Summary
5: Problem Solving Through Programming	20	Programming is used to efficiently solve problems. Relatively simple problems that require programming to efficiently obtain a solution are presented to students early in the course. For example, students are asked questions about images in terms of their color properties. Because the images contain thousands of pixels, computational thinking is required to answer the questions in a reasonable amount of time. Students are required to write programs that efficiently answer the questions, and they are challenged to pose questions that they and their peers can further explore. (CRD, DAT, AAP)
6: Guided Internet Explorations	15	Students explore the characteristics and uses of the internet as well as the World Wide Web (WWW). Protocols associated with the internet and the WWW are discussed. Security and privacy issues related to the use of the internet and the WWW are examined and discussed. For example, students simulate a binary transmission using small papers representing packets whose formats represent related protocols. They create and use Huffman coding trees to encode text messages into binary form, and then they model the messages using their packet models in paper form. Students also explore encryption and decryption techniques. (CRD, DAT, AAP, CSN)
7: Identifying Innovations That Connect Computing to People	15	Students select areas of interest that are impacted by computing. They research and report on those areas in the form of written and oral reports, as well as computational artifacts. Past activities have included a debate about the impact of computing on society, the creation of videos on computing in a variety of fields, and the development of infographics that illustrate computing information that is personally relevant. (CRD, IOC)
8: AP Performance Task: Create	12	Students have 9 hours of class time to complete the Create Performance Task and submit it to the College Board. (CRD, DAT, AAP) CR11

CR11
The syllabus must include an explicit statement that students are provided with at least nine (9) hours of class time to complete the AP Create Performance Task.

Activities Related to Each Exploration Theme:

Exploration Theme 0

Activity 1 – **Reading and Summarizing Articles About Computing** (CRD, IOC) (CTP5)

Students are given time to independently read the titles of a variety of articles highlighted on ACM TechNews. They work independently to select one article that they find the most interesting, read the article, and write a brief summary of the article in their journals. Students are then given an opportunity to share their insights and discoveries with their table partners and the entire class. As the year progresses, students are expected to improve their ability to use correct computer science terminology and to include details associated with computing innovations from the articles.

Activity 2 – **Creating and Using Piazza Accounts** (CRD, IOC) (CTP5)

Daily, students read and respond to posts created by their peers and the instructor. This is an ongoing formative assessment. Students also reflect on how the use of Piazza impacts their ability to obtain an education. Discussions led by the instructor include perspectives on how computing has impacted education over the last three decades. Students are given opportunities to explain how computing has impacted their personal educational experiences. CR3

CR3
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).

Exploration Theme 1

Activity 1 – **Introduction to Programming** (CRD, AAP) (CTP2)

Students utilize pair programming by working in pairs, one student as a "driver" and one student as a "navigator" to complete the "Introduction and Code" and "Digital Images" exercises on introcomputing.org plus extension exercises posted on Piazza. Traditional and enhanced for loops are introduced and applied to the creation of pixel patterns. The modulus operator % is defined and used to make patterns of pixels in rows, columns, and checkerboard patterns. Students use solid colors and pixels obtained from images to create the patterns. They use binary representations of data and binary operators to modify pixel colors. Students explore different base systems and complete problems involving conversions between bases. Discussions about manipulations of images are integrated on a regular basis in class, in small groups, and on Piazza. Students post screenshots of the output produced and the code they used to generate it. **CR3**

Activity 2 – **The Chaos Game** (CRD, AAP) (CTP2)

Students write code to implement a variety of versions of the chaos game. Multiple examples are given in the Creativity Curriculum Module, but students are encouraged to create images that are different from the examples. Differences may include variations such as color, vertex number and location, move to ratios, probabilities associated with the frequency of selecting particular vertices, and plotted pixel colors using images selected by students. They are encouraged to post at least three images to Piazza: one that was the result of collaborative work with a partner, one that was the result of individual work, and one that illustrates insights gained and the progress made during their individual explorations. Students post their images and corresponding code on Piazza, and comment and ask questions about their peers' posts.

Activity 3 – **Creativity and Computing** (CRD, AAP) (CTP2)

Students are given instructions outlining the characteristics of an image they are to create. The image creation process begins with an image provided by the instructor. Students write code to generate image modifications in the form of regular pixel patterns, color modifications, and fractals. In order to submit their final image, students must screen capture the image and submit both the screen capture and the code used to generate the final image. Students submit their summative assessments in the form of Piazza posts that are private to the instructor or as additions to their student journal.

Exploration Theme 2

Activity 1 – **The introcomputing.org JavaScript API** (CRD, DAT, AAP) (CTP2, CTP3)

Students identify the abstractions provided to them when using the introcomputing.org JavaScript programming website. Images, pixels, color values, tables, rows, and fields are some of the abstractions discussed related to the website. Students use these abstractions to create programs that create and manipulate images and data tables. Students create their own abstractions in the form of JavaScript functions that provide functionality related to programs that they find interesting related to images and table data.

Activity 2 – **Identifying and Describing Abstractions in Programming** (AAP, CTP3)

Students individually write a summary of an abstraction selected from a program provided by the instructor. The abstraction should correspond to code for some significant feature in the provided program. The summary must include the purpose of the abstraction, any input and output associated with the use of the abstraction, and any algorithms used by the abstraction to accomplish its related task. The summary must be written as a Piazza post that is private to the instructor.

Activity 3 – **Using Abstractions to Write Programs** (CRD, AAP) (CTP2, CTP3)

Students are given an API and provided examples of how to read it. A demonstration is provided of how a program feature is implemented by searching the API for programming abstractions that facilitate the implementation of the feature. Students read the related information about the abstraction, use the sample code provided in the API illustrating the use of the abstraction, and then add to and modify the sample code in a demo program. Students write program code in collaborative pairs for creative expression and problem solving. **CR6**

CR6
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.

Exploration Theme 3

Activity 1 – **Finding Appropriate Online Data Sources** (CRD, DAT, IOC) (CTP5, CTP6)

A variety of online data sources using the resources page on the course website are introduced. Students actively search the web in pairs (selected based on common interests) in order to find URLs for data sources that are relevant to topics of their choice. Students collect data from their chosen online sources using techniques that are appropriate for the data format that is available. Note that students may find several databases that are very difficult to use. They may also find it difficult to find freely available data sources associated with financial resources. Resources are provided that represent a compromise between individual interests and usable data sources.

Activity 2 – **Collecting, Cleaning, and Formatting Data** (CRD, DAT, AAP) (CTP2, CTP5)

The process of cleaning and formatting data in order to obtain new insights and knowledge is demonstrated. This cleaning often involves the removal of portions of the data that are not relevant to the area of interest. The cleaning and formatting of the data are accomplished using variety of tools such as a text editor, a spreadsheet, and JavaScript programs. In pairs, students discuss the techniques they will need to clean and format their data. Tasks are divided up to more efficiently prepare the data for information extraction. They apply the techniques discussed in class to their own data sets.

Activity 3 – **Data Trade-Offs** (CRD, DAT, IOC) (CTP5, CTP6)

Students are given questions about trade-offs associated with digital data (storage, security, privacy, compression) in a Google form for which the link is posted on Piazza. Students use class discussions, readings from *Blown to* Bits, and image manipulation exercises from introcomputing.org associated with binary representations to effectively answer questions using the form. The instructor provides a summary of all student answers as a Piazza post. Table groups read and discuss the summary. Each group member presents the ideas they support and those with which they disagree.

Exploration Theme 4

Activity 1 – **Comparing Algorithms for Image File Creation** (AAP) (CTP4)

Students are introduced to an algorithm to generate an image file. A second modified algorithm is presented that generates the same image file much more quickly. Students are asked to compare the two algorithms in terms of the number of steps needed to complete each algorithm and to test that the algorithms produce the same image. Discussions related to that and other algorithms lead students to an understanding of how to evaluate algorithms based on both correctness and efficiency. **CR3** **CR7**

CR7
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.

Activity 2 – **Creating Fractals for the Art Show** (CRD, AAP) (CTP1, CTP2, CTP3)

Students are formally introduced to a variety of fractal images whose creation rely on mathematical computations. Mathematical functions are explored, and the related fractals generated using those functions are discussed. Small changes in computations often result in dramatic changes in resulting images. Students design and create abstractions related to the geometry and color scheme of the images in order to create an image that can be shared with others in an art show. Software designed to create large, high-resolution versions of the images is presented and discussed. Students modify the software to create poster-size images of their designs.

Exploration Theme 5

Activity 1 – **Baby Name Analysis** (CRD, DAT, AAP) (CTP1, CTP2, CTP3)

Students independently write JavaScript code to answer the questions posed about baby names (Social Security Administration data base) in the Table Data section of introcomputing.org. They also pose name-related questions and answer their own questions and the questions of their peers using JavaScript programming. This programming can take place using the Brackets website or using text files that are loaded into a web browser. The W3Schools website is used as a reference for students who need to refresh their memory of JavaScript syntax and functionality. In small peer groups and in their weekly journal, using appropriate terminology and various levels of abstractions, students are required to explain how they solved the problems.

Activity 2 – **Problem Solving Through Programming** (CRD, DAT, AAP) (CTP1, CTP2)

Students work in pairs to write programs that solve problems posed on introcomputing.org. All solutions for these problems are written in the text areas provided within the site. Students must also pose problems to be solved by their table partners. Students write code to solve the problems posed by their peers using the website, Brackets, or simple text files that are loaded into the web browser. **CR5**

Activity 3 – **Mad Libs** (CRD, AAP) (CTP1, CTP6)

Students work in groups of four and create a story about a topic of interest. They then must create a webpage for the story containing images, colors, and general setting descriptions of their choosing. They must include input (text box, text area, checkbox, radio button, pull down menu, etc.) through which users will input nouns and verbs to be used in the story as determined by the group. Students brainstorm about how to divide the programming task among the group in order to use the strengths of each individual. They must write a JavaScript program within the webpage that uses input from the page to determine the final details of their story. Stories are shared by posting on Piazza. **CR4**

CR5
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.

CR4
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.

Exploration Theme 6

Activity 1 – **Simulating Internet Communications** (CRD, DAT, AAP, CSN) (CTP5)

Students search YouTube using the keywords “Huffman Coding Tree Encoding” and select videos that are effective for presenting encoding and decoding using Huffman coding trees (HCT). Each group posts their video URL choice on Piazza. An HCT is developed during class based on sample letter frequencies from instructor- and student-generated messages. The messages are translated into binary form on paper and then the message is broken into smaller sequences of 1s and 0s called packets. These packets are represented by small pieces of paper that are formatted using agreed-upon protocols. Some of the protocols include labeling the packets with source and destination information that ensures reassembly after transmission and packet requests of the source for those that are not received. Packets are passed through student “routers,” who receive packets and pass them toward their destination. Packets are reassembled at the message destination and translated from binary to the original message form. Students enjoy sending “secret messages” to their peers.

Activity 2 – **Understanding the Internet and the World Wide Web** (CSN) (CTP5, CTP6)

URLs associated with different iterations of the school website are posted on Piazza. Students are encouraged to view the history of the site and consider the persistent nature of data on the Internet and in particular, on the World Wide Web. Students work in table groups and generate a set of questions per table that they have about the internet. Each table posts their questions on Piazza: one post per unique question. Students use the internet, Chapter 7 and the Appendix from *Blown to Bits*, and the resources listed on the course website to research the answers to their peers' questions. Each table discusses and posts their best responses. Students respond to each other's comments, refining the answers and eventually resulting in fairly accurate responses to the original questions.

Activity 3 – **The Hardware and Binary Connection** (DAT, AAP) (CTP5, CTP6)

Students read the "Computer Hardware" page on introcomputing.org and explore the concepts (hardware, chip, CPU, transistor, etc.) and images (pictures of hardware) presented while examining actual hardware in the classroom. Students discuss the topics in table groups and answer questions about the hardware presented to them. Competition-style questions similar to those found in the American Computer Science League (ACSL) are introduced on the topics of Boolean algebra and logic gates. Sample problems are presented, discussed, and solved in large groups, small groups, and individually. Connections are made between the hardware used to create computing technology and binary systems. CR3

Activity 4 – **Creating Videos Explaining Characteristics of the Internet** (CRD, CSN) (CTP1, CTP5, CTP6)

The major characteristics of the internet are posted on the board (hierarchical, redundant, fault tolerant, standardized through protocols, packet-switched, and scalable) and students vote using a Piazza poll for the characteristic that is the most interesting to them. After participating in the poll, students move to a location in the room designated as representing the characteristic they selected and divide into teams. Students are instructed to create a video, no longer than one minute, which illustrates the characteristic of the internet that they voted for. When each team has completed their video, all submissions are edited into a single video illustrating the major characteristics of the internet. CR3

Exploration Theme 7

Activity 1 – **Identifying and Summarizing Articles About Computing Innovations and Impacts**

(IOC) (CTP5) (CI1, Prompts A and B)

Students browse the ACM TechNews website and select one article that identifies a computing innovation that impacts or has the potential to impact our society. Students write a summary of the article on Piazza and are given an opportunity to verbally share their insights with their table partners and with the entire class. Their summary must include a clear statement of how the innovation functions, what data it uses and how that data is produced or consumed, and how it impacts people and society (both beneficial and harmful impacts). Students post their summaries to Piazza. CR8 CR10

Activity 2 – **Wharton Top 30 Innovations List** (IOC) (CTP5) (CI2, Prompt C)

Students are given class time to read, in pairs, the list of the top 30 innovations from Wharton and select two computing innovations that are the most interesting to them. Students then write a technical summary as a Piazza post of how computing is related to those innovations. For at least one innovation, students must address the privacy and security of data related to that innovation. Students are then given an opportunity to share their insights with their table partners and the entire class. CR10

Activity 3 – **Debating the Beneficial and Harmful Impacts of Computing on Our Society** (CRD, IOC) (CTP6) (CI3, Prompt A)

Students stand and move to a portion of the room (Likert scale) that indicates the degree to which they believe a provided computing innovation has provided more benefits than harm to society. The teacher conducts a question and answer session that helps shift students in order to balance groups if needed. Each group selects one of the discussed

CR8
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.

CR10
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:

a. Explain beneficial and harmful effects of at least one computing innovation on society, economy, or culture.
b. 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.
c. Identify data privacy, security, or storage concerns for at least one computing innovation.

For each activity, use the label Computing Innovation 1, 2, or 3 to identify the activity, and use the label A, B, or C to identify the prompt(s).

computing innovations. Students are then given time in class to collaborate with their group, plan arguments for their side, and debate the beneficial and harmful impacts of their computing innovation. The debate is recorded and the resulting video summarizing the debate issues is edited and is presented to the class. CR9 CR10

CR9
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.

Exploration Theme 8

Activity 1 – **Create Performance Task** (CRD, DAT, AAP) (CTP1, CTP2, CTP3, CTP4)

Students complete the Create Performance Task as their final work before taking the AP Computer Science Principles Exam. At this point in the course, students are thoroughly familiar with the JavaScript programming. Students generally receive very positive feedback for their efforts in producing computational art for an art show, and they are highly motivated to continue the creation of personally relevant artifacts. They have worked almost an entire school year collaborating with their peers. At this stage in the course, they are comfortable creating both independently and collaboratively and independently summarizing their work.

Alignment to Topic Questions

Throughout each Exploration Theme, **Topic Questions** will be provided to help students check their understanding. The Topic Questions are especially useful for confirming understanding of difficult or foundational topics before moving on to new content or skills that build upon prior topics. Topic Questions can be assigned before, during, or after a lesson, and as in-class work or homework. Students will get rationales for each Topic Question that will help them understand why an answer is correct or incorrect, and their results will reveal misunderstandings to help them target the content and skills needed for additional practice.

Students will be assigned Topic Questions aligned to the learning objectives and skills in each Exploration Theme:

Exploration Theme	Big Ideas	Learning Objective and Skill	Topic Questions
0: Introduction to Computer Science Principles	IOC	IOC-1.A	None at this time.
1: Creativity and Computing	CRD	CRD-1.A, CRD-1.B, CRD-1.C	Topic 1.1
	DAT	DAT-1.A, DAT-1.B, DAT-1.C	Topic 2.1
	AAP	AAP-1.A, AAP-1.B	Topic 3.1
		AAP-2.A, AAP-2.B, AAP-2.C	Topic 3.3
		AAP-3.E.a, b	Topic 3.15
		AAP-3.F.a, b	Topic 3.16
2: Identifying, Creating, and Using Abstractions for Enhanced Creativity and Problem Solving	CRD	CRD-2.A, CRD-2.B, CRD-2.C, CRD-2.D	Topic 1.2
	AAP	AAP-3.A.a, b	Topic 3.12
		AAP-3.B, AAP-3.C	Topic 3.13
		AAP-3.D	Topic 3.14

Exploration Theme	Big Ideas	Learning Objective and Skill	Topic Questions
3: Using Data to Gather Information and Create New Knowledge	DAT	DAT-1.D	Topic 2.2
		DAT-2.A, DAT-2.B, DAT-2.C	Topic 2.3
	AAP	AAP-1.C, AAP-1.D	Topic 3.2
		AAP-2.N.a, b, AAP-2.O.a, b	Topic 3.10
		AAP-2.P.a, b,	Topic 3.11
	IOC	IOC-2.A, IOC-2.B, IOC-2.C	Topic 5.6
4: Exploring Algorithms for Enhanced Creativity and Problem Solving	CRD	CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H	Topic 1.3
		CRD-2.I.a, b, CRD-2.J	Topic 1.4
	AAP	AAP-2.L, AAP-2.M.a, b	Topic 3.9
		AAP-4.A.a, b	Topic 3.17
		AAP-4.B	Topic 3.18
5: Problem Solving Through Programming	DAT	DAT-2.D, DAT-2.E	Topic 2.4
	AAP	AAP-2.D	Topic 3.4
		AAP-2.E.a, b, AAP-2.F.a, b	Topic 3.5
		AAP-2.G, AAP-2.H.a, b	Topic 3.6
		AAP-2.I.a, b	Topic 3.7
		AAP-2.J, AAP-2.K.a, b	Topic 3.8
6: Guided Internet Explorations	CSN	CSN-1.A, CSN-1.B, CSN-1.C, CSN-1.D	Topic 4.1
		CSN-1.E.a, b, c	Topic 4.2
		CSN-2.A, a, b, CSN-2.B	Topic 4.3
7: Identifying Innovations That Connect Computing to People	IOC	IOC-1.A, IOC-1.B	Topic 5.1
		IOC-1.C	Topic 5.2
		IOC-1.D	Topic 5.3
		IOC-1.E	Topic 5.4
		IOC-1.F	Topic 5.5