
AP[®] Computer Science Principles

Sample Student Responses and Scoring Commentary

Inside:

Performance Task—Create

- Scoring Guideline**
- Scoring Commentary**

Student Samples provided separately

AP[®] COMPUTER SCIENCE PRINCIPLES — CREATE PERFORMANCE TASK

2019 SCORING GUIDELINES

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 1 Developing a Program with a Purpose	VIDEO & RESPONSE 2A	<ul style="list-style-type: none"> The video demonstrates the running of at least one feature of the program submitted. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do). 	<ul style="list-style-type: none"> Response earns the point if it explains the function of the program instead of identifying the purpose. Response earns the point if the illustrated feature runs, even if it does not function as intended. Response earns the point if the response is included in the video via narration or some form of closed captioning and addresses the purpose or function of the program. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> a video is not submitted; or the video does not illustrate the feature mentioned in the response; or the video does not illustrate the running of the feature (screen shots or storyboards are not acceptable and would not be credited). 	<ul style="list-style-type: none"> Purpose means the intended goal or objective of the program. Function means how the program works.
Row 2 Developing a Program with a Purpose	RESPONSE 2B	<ul style="list-style-type: none"> Describes or outlines steps used in the incremental and iterative development process to create the entire program. 	<p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the response only includes the process for determining the program idea and does not address the development process used to create the entire program; or the response does not indicate iterative development; or refinement and revision are not connected to feedback, testing, or reflection; or the response only describes the development at two specific points in time. 	<ul style="list-style-type: none"> Development processes are iterative and cyclical in nature and require students to reflect AND improve on what they have created. Examples of iterative development could include reflection, revision, testing and refining, and improvements based on feedback. The incremental and iterative development process does not need to be a formal method such as waterfall, top-down, bottom-up, agile, etc.
Row 3 Developing a Program with a Purpose	RESPONSE 2B	<ul style="list-style-type: none"> Specifically identifies at least two program development difficulties or opportunities. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> Describes how the two identified difficulties or opportunities are resolved or incorporated. 	<ul style="list-style-type: none"> Response earns the point if it identifies two opportunities, or two difficulties, or one opportunity and one difficulty AND describes how each is resolved or incorporated. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> only one distinct difficulty or opportunity in the process is identified and described; or the response does not describe how the difficulties or opportunities were resolved or incorporated. 	
Row 4 Applying Algorithms	CODE SEGMENT IN RESPONSE 2C	<ul style="list-style-type: none"> Selected code segment implements an algorithm. 	<p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the algorithm consists of a single instruction; or the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm). 	<ul style="list-style-type: none"> Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. (EU 4.1) Algorithms make use of sequencing, selection or iteration. (EK 4.1.1A)

AP[®] COMPUTER SCIENCE PRINCIPLES — CREATE PERFORMANCE TASK

2019 SCORING GUIDELINES

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 5 Applying Algorithms	RESPONSE 2C	<ul style="list-style-type: none"> • Selected code segment implements an algorithm that uses mathematical or logical concepts. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> • Explains how the selected algorithm functions. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> • Describes what the selected algorithm does in relation to the overall purpose of the program. 	<ul style="list-style-type: none"> • The algorithm being described can utilize existing language functionality or library calls. • Response earns the point even if the algorithm was not newly developed (i.e., a student’s reimplementation of the algorithm to find the minimum value). • Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • the selected algorithm does not include mathematical or logical concepts; • the response only describes what the selected algorithm does without explaining how it does it; or • the response does not explicitly address the program’s purpose; or • the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm). 	<ul style="list-style-type: none"> • See Row 4 definitions and curriculum framework alignment. • Mathematical concepts include mathematical expressions using arithmetic operators and mathematical functions. (EK 5.5.1.D) • Logical concepts include Boolean algebra and compound expressions. (EK 5.5.1E and 5.5.1F) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C)
Row 6 Applying Algorithms	RESPONSE 2C	<ul style="list-style-type: none"> • Selected code segment implements an algorithm that includes at least two or more algorithms. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> • At least one of the included algorithms uses mathematical or logical concepts. <p style="text-align: center;">AND</p> <ul style="list-style-type: none"> • Explains how one of the included algorithms functions independently. 	<ul style="list-style-type: none"> • Responses are still eligible to earn this row, even if they do not earn row 5. • The included algorithms can be sub-parts of the algorithm in row 5. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm). 	<ul style="list-style-type: none"> • See Row 4 and Row 5 definitions and curriculum framework alignment.

AP[®] COMPUTER SCIENCE PRINCIPLES — CREATE PERFORMANCE TASK

2019 SCORING GUIDELINES

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 7 Applying Abstraction	CODE SEGMENT IN RESPONSE 2D	<ul style="list-style-type: none"> Selected code segment is a student-developed abstraction. 	<ul style="list-style-type: none"> Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; or the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). 	<ul style="list-style-type: none"> The following are examples of abstractions (EK 5.3.1): <ul style="list-style-type: none"> ○ Procedures ○ Parameters ○ Lists ○ Application program interfaces (APIs) ○ Libraries Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1I)
Row 8 Applying Abstraction	RESPONSE 2D	<ul style="list-style-type: none"> Explains how the selected abstraction manages the complexity of the program. 	<ul style="list-style-type: none"> Responses should not be penalized for explanations of abstractions that are not developed by the student. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the explanation does not apply to the selected abstraction; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). 	<ul style="list-style-type: none"> See Row 7 definitions and curriculum framework alignment.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task

Note: Student samples are quoted verbatim and may contain spelling and grammatical errors.

Overview

Programming is a collaborative and creative process that brings ideas to life through the development of software. Programs can help solve problems, enable innovations, or express personal interests. In this performance task, students developed a program of their choice. The students' development process includes iteratively designing, implementing, and testing their program. Students were strongly encouraged to work with another student in their class.

Sample: A

Row 1: 1

Row 2: 1

Row 3: 1

Row 4: 1

Row 5: 1

Row 6: 1

Row 7: 1

Row 8: 1

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the purpose of the program is "to be an entertainment game and functions by requiring the user to collect 10 orbs using a spaceship to win."

Row 2:

The response earned the point for this row.

The response describes the incremental process of developing the program: "In the beginning I wanted to make a game about collecting items. First I assembled a list containing of the orbs that will be collected as the points variable. I then assigned arrow controls to the spider controlling the spaceship. I thought about setting it up so when the spaceship is within threshold of an orb, that orb will sink down and 1 point will be added. I also added a point limit that must be met to win. My game was too easy and it wouldn't decide whether a user wins or loses so I took the opportunity to add a timer. I did this by making a time variable and decrementing one off of it every second. The game is decided when the time equals zero." The iterative process is evident in the opportunity and difficulty descriptions.

Row 3:

The response earned the point for this row.

The response identifies an opportunity and a difficulty: "My game was too easy and it wouldn't decide whether a user wins or loses so I took the opportunity to add a timer. I did this by making a time variable and decrementing one off of it every second. The game is decided when the time equals zero. A difficulty I encountered while creating my program was how to restart just the game (not program). Everytime I pressed try again, the whole program freezes. To try to overcome this, I messed around with thee order of code until I solved it when I reset the variables to its original values because when I clicked try again, the conditions were still met so it looped."

AP[®] COMPUTER SCIENCE PRINCIPLES 2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 4:

The response earned the point for this row.
The code that is given represents an algorithm.

Row 5:

The response earned the point for this row.
The response includes an algorithm that has math/logic (if and while statements). The response explains how the world.Start function works: "My method world.Start functions by asking a yes or no question about if the user wants to play. Depending on the answer the spider will either be put into the spaceship (Yes) or put into free roam (No)." The response describes what the algorithm does in respect to the entire program: "world.Start is to be an introduction and a choice for the user if they want to play or not."

Row 6:

The response earned the point for this row.
The response gives an algorithm (world.Start) that uses two algorithms (world.Game and world.Finder). The response explains how world.Finder works: "uses math by making it so that when the jet is less than 5 meters from an orb from the orbs list, the orb will add a point and make the orbs disappear."

Row 7:

The response earned the point for this row.
The boxed code segment in the response represents a valid abstraction (a procedure or method).

Row 8:

The response earned the point for this row.
The response explains how the abstraction manages complexity. The description "when the game is restarted the code to make the game work the same way as when it was just started without having to be written again" is evidence that the function the student wrote is called from more than one place, which is a good rationale for managing complexity. Also, the comment "[t]his method hides all of these complex instructions that make the game replayable" is about managing complexity. The abstraction hides (i.e., encapsulates) the "complex" code that makes the game replayable. Finally, the response states that the name "WinnerandRestart" helps to explain "the function of the method."

Sample: B

Row 1: 1

Row 2: 1

Row 3: 1

Row 4: 1

Row 5: 1

Row 6: 1

Row 7: 1

Row 8: 1

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the program's purpose is "to create something fun and enjoyable for the person playing the game by having the player move a ship and dodge enemy ships."

Row 2:

The response earned the point for this row.

The response outlines steps in the development process: "first developed a collision system with a timed loop ... store the moves inside of a 2d array ... used a timed loop to scan each move."

Row 3:

The response earned the point for this row.

The response identifies one opportunity: "I realized that instead of having to make every single move by hand ... I could use an algorithm that checked how far away the player was." This is incorporated by writing an algorithm that "generated a move that moved it as close as possible to the player." The response also includes a difficulty: "When I needed a way to read each move and then moving the enemies over time to that location." This was resolved by using "a timed loop to scan each move and generate a new one when the first was done."

Row 4:

The response earned the point for this row.

The selected code segment provided is an algorithm, the `hit_timer_start()` function.

Row 5:

The response earned the point for this row.

The code segment identifies the function `hit_timer_start`. The code segment uses logic (if statements). The response describes how the algorithms work: "[a]fter my main algorithm uses 'collision_detector', if the player is touching an enemy it uses my second sub-algorithm 'kill.'" The response also describes what the selected algorithm does in relation to the overall purpose of the program: "sets the screen to 'dead_screen' where it sets the text of the labels to the score and time the player achieved while they were playing the game ... My main algorithm then checks if the user is touching the food and after sets the locations of the enemies and the players to where their coordinate variables say they should be. Together these algorithms add a way to kill the player. These help the purpose of the game by making it fun to play."

Row 6:

The response earned the point for this row.

The selected algorithm, `hit_timer_start()`, includes subalgorithms `collision_detector()` and `kill()`. The `collision_detector()` uses both math and logic. The subalgorithm `kill()` is fully described in the response.

Row 7:

The response earned the point for this row.

The selected abstraction is the student-created function `generate_move()`.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 8:

The response earned the point for this row.

The response discusses how the abstraction helps to manage complexity: "Therefore, the abstraction helps manage complexity because of the single function letting the algorithm be used multiple times. If I didn't use this abstraction, I would have to repeat the code 8 times which would make it more messy and harder to understand."

Sample: C**Row 1: 1****Row 2: 1****Row 3: 0****Row 4: 1****Row 5: 1****Row 6: 1****Row 7: 1****Row 8: 1****Row 1:**

The response earned the point for this row.

The video demonstrates the running of the program, and the purpose is given as "to simulate the game Snake, in which you play a snake which needs to get the targets in order to grow in size."

Row 2:

The response earned the point for this row.

The response describes the development process as "creating a sprite that moved ... make the sprite move in the direction that I wanted it to ... using the press key block and direction block ... brainstorming ideas and implementing them, such as creating a target and making the target move on impact." The iterative nature of development is also evident in the response: "After testing multiple times over, and revising the code, I finally fixed it."

Row 3:

The response did not earn the point for this row.

The response identifies only one difficulty and resolution: "Whilst trying to use the clone block to make my snake grow in size, it kept on touching itself, causing it to go to the game over screen," this was resolved by "changing the structure of my code by using a previous code segment that I had."

Row 4:

The response earned the point for this row.

The response identifies an algorithm.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 5:

The response earned the point for this row.

The algorithm contains math and logic (if statements, Boolean and arithmetic expressions). The response explains how the algorithm works: "first needs to set the target to appear and set the target to a certain size ... the target needs to go to a random position ... a repeat statement ... keeps running until it either touches the sprite, or till it goes off the screen. If it does touch the spirit, then a wait until the statement is completed." The response also describes what the purpose is for this algorithm in relation to the entire program: "to make the target move after touching my sprite, increase the difficulty of the game by changing the speed, and increase the length of my snake."

Row 6:

The response earned the point for this row.

The algorithm uses two algorithms that run independently and have been explicitly identified: the block "go to random position" ("through a custom block that I incorporated") and the forever loop ("Once it has gone to that position, a repeat statement starts to run"). The algorithm in the forever loop contains logic via if-statements, repeats, waits, and Boolean expressions. The response also explains how that algorithm functions independently. The parent algorithm is the entire included code, which is triggered by "When green flag clicked." Because the second subalgorithm is included inline, its description blends with the description of the main algorithm, but it is still counted as a subalgorithm.

Row 7:

The response earned the point for this row.

The given function "RUN" is a student-created abstraction.

Row 8:

The response earned the point for this row.

The response explains how complexity is managed: "allowing for an ease to understand and debug my script ... With this abstraction implemented, the process of debugging also becomes easier, as if something is wrong with how my snake sprite is working in the game, I could instantly know to look at the custom block first to see if everything inside of the block is in working order."

Sample: D**Row 1: 1****Row 2: 1****Row 3: 1****Row 4: 1****Row 5: 0****Row 6: 1****Row 7: 1****Row 8: 1**

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the purpose of the program is "to create something fun that can interact with the player as well as provide a challenge to the user."

Row 2:

The response earned the point for this row.

The response describes the incremental process in developing the program: "I wrote program firstly by giving movement to the basket using arrow keys as well as allowing the apples to drop from the sky to be caught by the basket which would then either give a point for a score if caught or subtract a point for lives if missed." The iterative process is evident in the difficulty and problem descriptions.

Row 3:

The response earned the point for this row.

The response identifies a difficulty and a problem and includes how they were resolved.

Row 4:

The response earned the point for this row.

The code that is given represents an algorithm.

Row 5:

The response did not earn the point for this row.

The response does not identify clearly which is the "main" algorithm; there is no relation to overall purpose for the Game function, which seems to be the main algorithm based on the paragraph's first sentence; and there is no explanation at all of the Game function. Instead, the response focuses on the subalgorithms, which are otherwise unrelated.

Row 6:

The response earned the point for this row.

The response gives an algorithm (Game) that uses two algorithms (MoveBasket and moveApple).

Row 7:

The response earned the point for this row.

The boxed code segment in the response represents a valid abstraction (function).

Row 8:

The response earned the point for this row.

The response explains how the abstraction manages complexity: "My code utilizes abstraction by involving a function to repeat itself several times to manage complexity and make the code easier to understand."

Additionally, it further explains that "[t]his abstraction makes my program more manageable by taking a part of the code that would have repeated itself multiple times and condensing it to reduce complexity."

AP[®] COMPUTER SCIENCE PRINCIPLES 2019 SCORING COMMENTARY

Create Performance Task (continued)

Sample: E

Row 1: 1

Row 2: 0

Row 3: 1

Row 4: 1

Row 5: 1

Row 6: 0

Row 7: 1

Row 8: 1

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the program's purpose is "to guess which option the computer is going to randomly input."

Row 2:

The response did not earn the point for this row.

The does not describe or outline steps used in the development process.

Row 3:

The response earned the point for this row.

The response identifies one difficulty: "what to name the pictures so that they can be randomly picked by the computer." This was resolved by assigning "each picture a number and then used randomNumber so the computer could have a randomly assigned picture." The response also mentions a second difficulty: "the score would not display." This was resolved by realizing that "the text box was not large enough for the number to display."

Row 4:

The response earned the point for this row.

The selected code segment provided is an algorithm, `checkCorrect(buttonId)`.

Row 5:

The response earned the point for this row.

The written response describes the function `checkCorrect`, which "checks to see if the user input matches the random computer number." The code segment uses logic (if statements). The response explains how the algorithms work: "The '`checkCorrect`' algorithm checks to see if the user input matches the random computer number ... If the inputs match, the score is updated to add 1 and the '`resultLabel`' displays 'correct.' If the inputs don't match. "

Row 6:

The response did not earn the point for this row.

The included algorithm includes subalgorithms `updateScore` and `endGame`. However, `endGame` is never called from the selected "parent" algorithm.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 7:

The response earned the point for this row.
The included abstraction is a student-created function.

Row 8:

The response earned the point for this row.
The response discusses that complexity is managed by writing the function once and calling it many times.

Sample: F**Row 1: 1****Row 2: 0****Row 3: 1****Row 4: 1****Row 5: 1****Row 6: 0****Row 7: 0****Row 8: 0****Row 1:**

The response earned the point for this row.
The video shows the program running, and the response identifies the purpose as "a random terrain generator."

Row 2:

The response did not earn the point for this row.
The response shows indirect evidence of iterative development in testing and fixing of bugs, but there is no discussion of an incremental development process and no explicit description of a development process.

Row 3:

The response earned the point for this row.
The response identifies a difficulty: "determining the variables and how they were going to play into the program." This difficulty was resolved by using "a variable to determine the randomly generated height of each column ... to clone the sprite." A second difficulty is also identified: "I could not create layers of grass, dirt, and stone because the clone could only maintain one costume." This second difficulty was resolved by "removing the clone function and inserting a stamp function instead."

Row 4:

The response earned the point for this row.
The selected code is an algorithm.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 5:

The response earned the point for this row.

The selected algorithm uses logic (Boolean conditions and random numbers). The response describes what the selected algorithm does in relation to the overall purpose of the program: "complete the purpose of the program to create a randomly generated terrain." The response also explains how the selected algorithm functions: "repeat until the sprite hits the X position of 240, or the end of the stage. Each time ... the X position of the sprite is increased by 8, eventually reaching 240 so the code will stop repeating."

Row 6:

The response did not earn the point for this row.

The response does not identify two subalgorithms included in the selected algorithm.

Row 7:

The response did not earn the point for this row.

The selected code in the response is a control structure, not a student-created abstraction.

Row 8:

The response did not earn the point for this row.

The response does not discuss how the abstraction manages the complexity of the program.

Sample: G**Row 1: 1****Row 2: 1****Row 3: 0****Row 4: 1****Row 5: 1****Row 6: 1****Row 7: 1****Row 8: 0****Row 1:**

The response earned the point for this row.

The video demonstrates the running of the program, and the written response indicates the program's purpose is to be "a visual memory game."

Row 2:

The response earned the point for this row.

The response outlines the steps of the process in parts (including the final part, which is called out as an opportunity) to illustrate the incremental nature of their development process. The iterative nature of the development process is highlighted through a difficulty with resolution.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 3:

The response did not earn the point for this row.

The response only identifies one difficulty: "get the tiles to appear beyond a 3x3." This was resolved by remembering "to add an ending part to the define." The response also mentions the opportunity "to develop the clicker function," but this is an experiential opportunity, not an opportunity in program development.

Row 4:

The response earned the point for this row.

The selected code segment provided is an algorithm, `tock()`.

Row 5:

The response earned the point for this row.

The code segment identifies the function `tock()`, which contains logic in the form of if functions. The response explains how the function `tock()` functions: "It first checks to see if the game is in the playing state. If so, it determines whether the user has remaining lives to keep going. If they do, the game board with the white tiles is displayed." The response also describes what the selected algorithm does in relation to the overall purpose of the program: "The function implements the rules of the game."

Row 6:

The response earned the point for this row.

The included algorithm uses the subalgorithms `ready-to-close-open-tiles`, `advance-to-next`, `next-playing`, `reverse-to-previous`, and `previous-playing`. The subalgorithm `ext-playing` is fully described.

Row 7:

The response earned the point for this row.

The included abstraction is a student-created record structure, `PlayingStruct`.

Row 8:

The response did not earn the point for this row.

While the response states that the data structure is used in a number of functions, it does not explain how this helps to manage complexity.

Sample: H**Row 1: 1****Row 2: 0****Row 3: 1****Row 4: 1****Row 5: 1****Row 6: 0****Row 7: 0****Row 8: 0**

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program. The response provides the purpose of the program: "a game where the user clicks pictures of comets to get points within a time limit."

Row 2:

The response did not earn the point for this row.

The response does not address the development process.

Row 3:

The response earned the point for this row.

The response identifies a difficulty in finding "which numbers to use for the comets ... to make sure none of the comets ever went off the screen," which was resolved by using "the highest and lowest numbers for both the x and the y coordinates." The response identifies a second difficulty, in that the comets and the clock were both based on the same timer, and this broke when the student tried to make the comets move faster. This was resolved by adding "a second clock, and separated the timer from the comets."

Row 4:

The response earned the point for this row.

The response identifies the when Clock1.timer algorithm.

Row 5:

The response earned the point for this row.

The algorithm incorporates logic (if statements) and math (increment). It also explains how the algorithm functions: "the time variable decreases by one ... the timer label is changed." The response also describes what the algorithm does in relation to the program: "the time variable counts down every second, which is visible to the user of the app, and once it hits zero, another screen is opened to show the user the result of the game."

Row 6:

The response did not earn the point for this row.

The response does not identify any subalgorithms.

Row 7:

The response did not earn the point for this row.

The response identifies a variable as the abstraction. Variables are explicitly excluded as student-defined abstractions.

Row 8:

The response did not earn the point for this row.

The response does not address managing program complexity.

AP[®] COMPUTER SCIENCE PRINCIPLES 2019 SCORING COMMENTARY

Create Performance Task (continued)

Sample: I

Row 1: 1

Row 2: 0

Row 3: 1

Row 4: 1

Row 5: 0

Row 6: 0

Row 7: 0

Row 8: 0

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the purpose of the program is "to let the user experience a fun roleplay adventure by completing different mini games."

Row 2:

The response did not earn the point for this row.

The response does not describe or outline steps used in the incremental development process to create the entire program.

Row 3:

The response earned the point for this row.

The first difficulty is that "images ... ended up being squished and distorted," and this was resolved by choosing "a much smaller picture." The second difficulty is "making all the barriers to function" (restarting the maze), and this was resolved by naming "each individual" barrier differently, "[s]ame as for having the coins to be collected in the maze."

Row 4:

The response earned the point for this row.

The response identifies an algorithm. Note that the algorithm consists of several different `onClick()` event handlers, each of which merely calls `setScreen()` to switch to a different screen. Although this may not appear to be an algorithm because there is only one line inside each event handler, the collection of event handlers as a whole is an algorithm because algorithms are language-independent, and this would correspond to a large `if-else-if` or `switch` statement in other programming languages.

Row 5:

The response did not earn the point for this row.

The response does not indicate how the algorithm works or what it does. The written response is incomplete.

Row 6:

The response did not earn the point for this row.

The response does not include two or more subalgorithms. The written response is incomplete.

AP[®] COMPUTER SCIENCE PRINCIPLES 2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 7:

The response did not earn the point for this row.

The response identifies an event handler as the abstraction, and this is not a student-created abstraction.

Row 8:

The response did not earn the point for this row.

The response does not address managing program complexity. The written response is incomplete.

Sample: J

Row 1: 1

Row 2: 0

Row 3: 0

Row 4: 1

Row 5: 0

Row 6: 0

Row 7: 0

Row 8: 0

Row 1:

The response earned the point for this row.

The video demonstrates the running of the program, and the response indicates how the program functions.

Row 2:

The response did not earn the point for this row.

The response does not describe the development process of the entire program in an incremental or iterative manner.

Row 3:

The response did not earn the point for this row.

The response only identifies one difficulty related to the use of the math block.

Row 4:

The response earned the point for this row.

The selected code segment provided is an algorithm, favoriteyoutuber.

Row 5:

The response did not earn the point for this row.

The code segment identified in the response uses logic (if statements), and the response explains how the algorithm works: "it asks 'Who's your favorite youtuber?' on every question it says 'It's {youtubername!} That's such a good choice!'" But the response does not describe what the algorithm does in relation to the overall purpose of the program.

AP[®] COMPUTER SCIENCE PRINCIPLES

2019 SCORING COMMENTARY

Create Performance Task (continued)

Row 6:

The response did not earn the point for this row.

The code segment that is included in the response does not include an algorithm that incorporates two additional algorithms. There are two different onEvents, but they are not incorporated in a main "parent" algorithm.

Row 7:

The response did not earn the point for this row.

The code segment that is included in the response is not a student-defined abstraction. An event handler is considered a built-in abstraction in this language.

Row 8:

The response did not earn the point for this row.

The response does not explain how the abstraction manages complexity for the program.