Create Performance Task Scoring Guidelines

- There are **six rows** (0-1 point):
  - Program Purpose and Function
  - Data Abstraction
  - Managing Complexity
  - Procedural Abstraction
  - Algorithm Implementation
  - Testing

- Except where otherwise noted, each point of the rubric is earned independently. For instance, a student could earn a point for **Algorithm Implementation** without earning the point for **Procedural Abstraction**.

- Responses should be evaluated on the rationale provided not on the interpretation or inference on the part of the scorer.

- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria. You will find the terminology list on the last page.

AP Computer Science Principles 2021 Scoring Guidelines

Create Performance Task — 6 points

Learning Objectives: CRD-2.B AAP-1.D.a AAP-1.D.b AAP-3.C AAP-2.H.a AAP-2.K.a CRD-2.J

General Scoring Notes
- Responses should be evaluated on the rationale provided not on the interpretation or inference on the part of the scorer.
- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria.

Reporting Category	Scoring Criteria	Decision Rules
Row 1 Program Purpose and Function (0-1 points) 4.A	The video demonstrates the running of the program including: • *input*; AND • *program functionality*; AND • *output* AND The written response: • describes the overall *purpose* of the program; AND • describes what functionality of the program is demonstrated in the video; AND • describes the input and output of the program demonstrated in the video.	Consider ONLY the video and written response 3a when scoring this point. Do NOT award a point if any one or more of the following is true: • the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)

Row 2 Data Abstraction (0-1 points) 3.B	The written response:
	• includes two *program code segments*:
	  • one that shows how *data has been stored in this list* (or other *collection type*); and
	  • one that shows the data in this same *list being used* as part of fulfilling the program's purpose;
	AND
	• identifies the name of the variable representing the list being used in this response; AND
	• describes what the data contained in this list is representing in the program.
		Consider ONLY written response 3b when scoring this point.
		
		The written response must include two clearly distinguishable program code segments, but these segments may be disjointed code segments or two parts of a contiguous code segment.
		
		If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.
		
		Do NOT award a point if any one or more of the following is true:
		• the use of the list is trivial and does not assist in fulfilling the program's purpose.

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules
Row 1 **Program Purpose and Function** (0-1 points) 4.A	The video demonstrates the running of the program including: • *input*; AND • *program functionality*; AND • *output* **AND** The written response: • describes the overall *purpose* of the program; **AND** • describes what functionality of the program is demonstrated in the video; **AND** • describes the input and output of the program demonstrated in the video.	Consider ONLY the video and written response 3a when scoring this point. **Do NOT award a point if any one or more of the following is true:** • the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)

- The following are found in the terminology:
  - Input
  - Program Functionality
  - Output
  - Purpose

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules

Row 2 **Data Abstraction** **(0-1 points)** **3.B**	The written response:
	• includes two *program code segments*:
	  • one that shows how *data has been stored in this list* (or other *collection type*); and
	  • one that shows the data in this same *list being used* as part of fulfilling the program's purpose;
	AND
	• identifies the name of the variable representing the list being used in this response;
	AND
	• describes what the data contained in this list is representing in the program.
		Consider ONLY written response 3b when scoring this point.
		
		The written response must include two clearly distinguishable program code segments, but these segments may be disjointed code segments or two parts of a contiguous code segment.
		
		If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.
		
		Do NOT award a point if any one or more of the following is true:
		• the use of the list is trivial and does not assist in fulfilling the program's purpose.

- The following are found in the terminology:
  - Program Code Segment
  - List
  - Data has been stored in this list
  - Collection Type
  - List being used

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules
Row 3 Managing Complexity (0-1 points) 3.C	The written response: • includes a program code segment that shows a list being used to manage complexity in the program; AND • explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	Consider ONLY written response 3b when scoring this point. Responses that do not earn row 2, may still earn this row. Do NOT award a point if any one or more of the following is true: • the code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list.); OR • the written response does not name the selected list (or other collection type); OR • the use of the list is irrelevant or not used in the program; OR • the explanation does not apply to the selected list; OR • the explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program; OR • the solution without the list is implausible, inaccurate, or inconsistent with the program; OR • the use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier; OR • the use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules

Row 4 Procedural Abstraction (0-1 points) 3.B	The written response:
	• includes two program code segments:
	  • one of a *student-developed procedure*. The procedure:
	    - contains at least one *parameter*; and
	    - the parameter has an effect on the functionality of the procedure;
	  • one showing where the student-developed procedure is being called;
	AND
	• describes what the identified procedure does and how it contributes to the overall functionality of the program.
		Consider ONLY written response 3c when scoring this point.
		
		The procedure must be student developed, but could be developed collaboratively with a partner.
		
		If multiple procedures are included, use the first procedure to determine whether the point is earned.
		
		Do NOT award a point if any one or more of the following is true:
		• the code segment consisting of the procedure is not included in the written responses section; OR
		• the procedure is a built-in or existing procedure or language structure, such as an event handler or main method, where the student only implements the body of the procedure rather than defining the name, return type (if applicable) and parameters; OR
		• the written response describes what the procedure does independently without relating it to the overall function of the program.

• The following are found in the terminology:
  • Student-developed Procedure / Algorithm
  • Procedure
  • Parameter
  • Arguments

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules
Row 5 Algorithm Implementation **(0-1 points)** **2.B**	The written response: • includes a program code segment of a *student-developed algorithm* that includes: • *sequencing*; AND • *selection*; AND • *iteration*; **AND** • explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	Consider ONLY written response 3c when scoring this point. Responses that do not earn row 4 may still earn this row. The algorithm being described can utilize existing language functionality or library calls. An algorithm that contains selection and iteration, also contains sequencing. An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. Use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. Do **NOT** award a point if any one or more of the following is true: • the response only describes what the selected algorithm does without explaining how it does it; OR • the description of the algorithm does not match the included program code; OR • the code segment consisting of the selected algorithm is not included in the written response; 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); OR • the use of either the selection or the iteration is trivial and does not affect the outcome of the program.

The following are found in the terminology:
Sequencing
Selection
Iteration

Create Performance Task Scoring Guidelines *cont.*

Reporting Category	Scoring Criteria	Decision Rules
Row 6 Testing (0-1 points) 1.B	The written response: • describe two calls to the selected procedure identified in written response 3c. Each call must pass a different *argument(s)* that causes a different segment of code in the algorithm to execute; **AND** • describes the condition(s) being tested by each call to the procedure; **AND** • identifies the result of each call.	Consider **ONLY the written response for 3d and the selected procedure identified in written response 3c.** Responses that do not earn row 4 may still earn this row. **Do NOT award a point if any one or more of the following is true:** • a procedure is not identified in written response 3c or the procedure does not have a parameter; OR • the written response for 3d does not apply to the procedure in 3c; OR • the two calls cause the same segment of code in the algorithm to execute even if the result is different; OR • the response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program; OR • the identified results of either call are implausible, inaccurate, or inconsistent with the program.