2018

AP Computer Science Principles

Scoring Guidelines

AP Central is the official online home for the AP Program: apcentral.collegeboard.org

AP® Computer Science Principles — Create Performance Task
2018 Scoring Guidelines and Notes

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 1 **Developing a Program with a Purpose**	**VIDEO & RESPONSE 2A**	• The video demonstrates the running of at least one feature of the program submitted. **AND** • The response (audio narration or written response) identifies the purpose of the program (what the program is attempting to do).	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 video includes a narration or some form of closed captioning that addresses the purpose of the program. **Do NOT award a point if any one of the following is true:** ● a video is not submitted; ● 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).	● 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**	• Describes or outlines steps used in the incremental and iterative development process to create the entire program.	**Do NOT award a point if any one of the following is true:** ● the response does not indicate iterative development; ● refinement and revision are not connected to feedback, testing, or reflection; or ● the response only describes the development at two specific points in time.	● 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**	• Specifically identifies at least two program development difficulties or opportunities. **AND** • Describes how the two identified difficulties or opportunities are resolved or incorporated.	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. **Do NOT award a point if any one of the following is true:** ● 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**	• Selected code segment implements an algorithm.	**Do NOT award a point if any one of the following is true:** ● the algorithm consists of a single instruction; ● 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).	● 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)

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 5 **Applying Algorithms**	**RESPONSE 2C**	• Selected code segment implements an algorithm that uses mathematical or logical concepts. **AND** • Explains how the selected algorithm functions. **AND** • Describes what the selected algorithm does in relation to the overall purpose of the program.	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) **Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; ● the selected algorithm consists solely of library calls to existing language functionality; ● 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; ● the response does not explicitly address the program's purpose; ● 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).	● 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**	• Selected code segment implements an algorithm that includes at least two or more algorithms. **AND** • At least one of the included algorithms uses mathematical or logical concepts. **AND** • Explains how one of the included algorithms functions independently.	**Do NOT award a point if any one of the following is true:** ● the selected algorithm consists of a single instruction; ● the selected algorithm consists solely of library calls to existing language functionality; ● neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; ● 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).	● See Row 4 and Row 5 definitions and curriculum framework alignment.
Row 7 **Applying Abstraction**	**CODE SEGMENT IN RESPONSE 2D**	• Selected code segment is a student-developed abstraction.	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. **Do NOT award a point if any one of the following is true:** ● the response is an ***existing*** abstraction such as variables, existing control structures, event handlers, APIs; ● 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).	● The following are examples of abstractions (EK 5.3.1): o Procedures o Parameters o Lists o Application program interfaces (APIs) o 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**	• Explains how the selected abstraction manages the complexity of the program.	Responses should not be penalized for explanations of abstractions that are not developed by the student. **Do NOT award a point if any one of the following is true:** ● 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).	● See Row 7 definitions and curriculum framework alignment.