

2026



AP[®] Computer Science Principles

Scoring Guidelines

© 2026 College Board. College Board, Advanced Placement, AP, AP Central, and the acorn logo are registered trademarks of College Board. Visit College Board on the web: collegeboard.org.

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

Video, Program Requirements, and Written Response 1

3 points

General Scoring Notes

- Written responses should be evaluated solely on the rationale provided.
- Responses must demonstrate all scoring criteria, including those within bulleted lists, in each reporting category to earn the point for that category.
- Terms and phrases defined in the terminology list are italicized when they first appear.

Reporting Category	Scoring Criteria	Decision Rules
Course Project: Video (0–1 points)	The video demonstrates the running of the program including: • <i>input</i> • <i>program functionality</i> • <i>output</i>	Consider the video (or Program Code if necessary) when scoring this point. • The video needs to show at least one example of the program’s functionality. • If the source of the input is unclear from the video, consider the full program code file when scoring this point. Do NOT award a point if 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).
Course Project: Program Requirements (0–1 points)	The program code includes: • <i>A student-developed procedure</i> • <i>A call to the student-developed procedure</i> • <i>A list (or other collection type)</i> • <i>A use of the list</i> • <i>Selection</i> • <i>Iteration</i>	Consider the Personalized Project Reference (or Program Code if necessary) when scoring this point. • If the program requirements do not appear in the Personalized Project Reference, consider the full program code file when scoring this point. • The procedure does not need to have a <i>parameter</i> to earn this point. • The <i>code segments</i> demonstrating selection and iteration do not need to appear in the same <i>algorithm</i> to earn this point. • The code segments demonstrating selection and iteration do not need to be contained in a procedure to earn this point. Do NOT award a point if any one or more of the following is true: • The list is a one-element list. • The use of the list is irrelevant (i.e., not connected to the program’s functionality). • The call to the procedure is inconsistent with the procedure header (unless allowed by the programming language). • The use of either the selection or the iteration is trivial (i.e., does not affect the outcome of the program).

Reporting Category	Scoring Criteria	Decision Rules
Written Response 1: Program Design, Function, and Purpose (0–1 points)	The written response: - describes a piece of documentation that would be appropriate to include with or in the program. - explains how the documentation can be used by another programmer to better understand a particular code segment in the program.	Consider the Video (or Program Code if necessary) and Written Response 1 when scoring this point. - Documentation is a description (e.g., text, video, or diagram) that helps someone understand how a code segment, procedure, or program functions or was developed. - Appropriate uses of documentation include acknowledgment of code segments written by someone else or descriptions of what a code segment does or how it works. - Responses can be a general description of the documentation or the actual documentation (e.g., a specific comment) that would be included with or in the program code. Do NOT award a point if any one or more of the following is true: - The description of the documentation is implausible, inaccurate, or inconsistent with the program. - The explanation only references the user of the program and not another programmer.

Written Response 2

3 points

General Scoring Notes

- Written responses should be evaluated solely on the rationale provided.
- Written responses must demonstrate all scoring criteria, including those within bulleted lists, in each reporting category to earn the point for that category.
- Terms and phrases defined in the terminology list are italicized when they first appear.

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(a): Algorithm Development (0–1 points)	The written response: • identifies the variables and specific value(s) that cause the <i>iteration</i> statement to stop executing. • explains why the identified value(s) cause the iteration statement to stop. OR • explains how the iteration statement stops executing if stopping the iteration statement is not controlled by any variables or specific values.	Consider the Personalized Project Reference and Written Response 2(a) when scoring this point. • If multiple iteration statements are included in the Procedure section of the Personalized Project Reference (part (i) or (ii)), use the first iteration statement to determine whether the point is earned. • The iteration statement does not need to be contained in a <i>procedure</i> to earn this point. • The response can earn the point if the identified values lead to the program crashing as a way of exiting the loop, including forever loops. • The response can earn the point if it describes how specific values are obtained or derived from the code even if the specific values are not explicitly given (e.g., using the length of a <i>list</i> as an identified value). Do NOT award a point if any one or more of the following is true: • No iteration statement appears in the Procedure section of the Personalized Project Reference. • The identified value(s), if given, do not cause the iteration statement to stop. • The explanation is inconsistent with the code in the first iteration statement. • The use of the iteration is trivial (i.e., does not affect the outcome of the program).

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(b): Errors and Testing (0–1 points)	The written response: - includes a call to the procedure with specific <i>argument</i>(s) that will cause the procedure to behave incorrectly. - describes the incorrect behavior. - explains why the incorrect behavior occurs as a result of the call. OR - explains why it is not possible for a call with accepted arguments to cause the procedure to behave incorrectly.	Consider the Personalized Project Reference and Written Response 2(b) when scoring this point. - If multiple procedures are included in part (i) of the Procedure section of the Personalized Project Reference: - Use the procedure referenced in the written response to determine whether the point is earned. - If no procedure is referenced in the written response, then use the first procedure to determine whether the point is earned. - The procedure must have at least one explicit <i>parameter</i>. Explicit parameters are defined in the header of the procedure. - The described behavior must be related to or dependent on the explicit parameter(s). - A response that describes only the part of the procedure’s behavior that is relevant to the prompt may still earn this point. - The call may be the same as the one in the Personalized Project Reference. - The syntax of the procedure call does not need to be correct if the response identifies the correct arguments to the procedure. - A description of the call rather than program code is acceptable. Do NOT award a point if any one or more of the following is true: - A procedure is not identified in part (i) of the Procedure section of the Personalized Project Reference. - The response does not apply to the procedure in part (i) of the Procedure section of the Personalized Project Reference. - The use of the explicit parameter is irrelevant (e.g., does not affect the <i>code segment</i> of the procedure that is executed or is reassigned before being used). - The identified arguments are not given specific values and are only described generally. - The response includes an explanation that is implausible, inaccurate, or inconsistent with the procedure and its given argument(s). - The response explains why it is not possible to write a call with arguments that cause the procedure to behave incorrectly instead of explaining why it is not possible for a call with accepted arguments to cause the procedure to behave incorrectly.

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(c): Data and Procedural Abstraction (0–1 points)	The written response: - explains how the list uses abstraction to manage complexity in the program. AND - describes how the code segment in part (ii) of the List section would have to be adjusted to have the same behavior without the list. OR - explains why it is not possible to have the same behavior without the use of the list.	Consider the Personalized Project Reference and Written Response 2(c) when scoring this point. - If multiple lists are included in the List section of the Personalized Project Reference: - Use the list(s) referenced in the written response to determine whether the point is earned. - If no list is referenced in the response, then use the first list to determine whether the point is earned. Do NOT award a point if any one or more of the following is true: - A list (or other <i>collection type</i>) is not included in the List section of the Personalized Project Reference. - The <i>list</i> is not <i>being used</i> in part (ii) of the List section of the Personalized Project Reference. - The response includes an explanation or describes a behavior that is implausible, inaccurate, or inconsistent with the program.

AP Computer Science Principles Create Performance Task Terminology

Algorithm: An algorithm is a finite set of instructions that accomplish a specific task. Every algorithm can be constructed using combinations of sequencing, selection, and iteration.

Arguments: The values of the parameters when a procedure is called.

Code segment: A code segment refers to a collection of program statements that are part of a program. For text-based, the collection of program statements should be continuous and within the same procedure. For block-based, the collection of program statements should be contained in the same starter block or what is referred to as a “Hat” block.

Collection type: Aggregates elements in a single structure. Some examples include: databases, hash tables, dictionaries, sets, or any other type that aggregates elements in a single structure.

Data stored in a list: Input into the list can be through an initialization or through some computation on other variables or list elements.

Input: Program input is data that are sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile (through touch), audible, visual, or text. An event is associated with an action and supplies input data to a program.

Iteration: Iteration is a repetitive portion of an algorithm. Iteration repeats until a given condition is met or for a specified number of times. The use of recursion is a form of iteration.

List: A list is an ordered sequence of elements. The use of lists allows multiple related items to be represented using a single variable. Lists are referred to by different terms, such as arrays or arraylists, depending on the programming language.

List being used: Using a list means the program is creating new data from existing data or accessing multiple elements in the list.

Output: Program output is any data that are sent from a program to a device. Program output can come in a variety of forms, such as tactile, audible, visual, movement, or text.

Parameter: A parameter is an input variable of a procedure. Explicit parameters are defined in the procedure header. Implicit parameters are those that are assigned in anticipation of a call to the procedure. For example, an implicit parameter can be set through interaction with a graphical user interface.

Procedure: A procedure is a named group of programming instructions that may have parameters and return values. Procedures are referred to by different names, such as method, function, or constructor, depending on the programming language. A procedure is executed through the use of a procedure call.

Program functionality: The behavior of a program during execution, often described by how a user interacts with it.

Purpose: The problem being solved or creative interest being pursued through the program.

Selection / conditional statement: A selection / conditional statement affects the sequential flow of control by executing different statements based on a condition being true or false. The use of if-statements and try / exception statements are examples of selection / conditional statements.

Sequencing: The application of each step of an algorithm in the order in which the code statements are given.

Student-developed procedure / algorithm: Program code that is student-developed has been written (individually or collaboratively) by the student who submitted the response. Calls to existing program code or libraries can be included but are not considered student-developed. Event handlers are built-in abstractions in some languages and will therefore not be considered student-developed. In some block-based programming languages, event handlers begin with “when”.