

Written Response 2**3 points****General Scoring Notes**

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

Reporting Category	Scoring Criteria	Decision Rules
<p>Written Response 2(a): Algorithm Development</p> <p>(0–1 points)</p>	<p>The written response:</p> <ul style="list-style-type: none"> • identifies the number of times the body of the iteration statement will execute. <p>AND</p> <ul style="list-style-type: none"> ○ describes a condition or error that would cause an infinite loop. <p>OR</p> <ul style="list-style-type: none"> ○ if no such condition or error exists, explains how the loop could be modified to cause an infinite loop. 	<p>Consider the Personalized Project Reference and Written Response 2(a) when scoring this point.</p> <ul style="list-style-type: none"> • If multiple iteration statements are included in the Procedure section of the Personalized Project Reference, use the first iteration statement to determine whether the point is earned. • The iteration statement does not need to be contained in a procedure to earn this point. <p>Do NOT award a point if the following is true:</p> <ul style="list-style-type: none"> • The identified number of times the body of the iteration statement will execute does not match the code.
<p>Written Response 2(b): Errors and Testing</p> <p>(0–1 points)</p>	<p>The written response:</p> <ul style="list-style-type: none"> • describes a change to the procedure that will result in a run-time error. • explains why the change will result in a run-time error. 	<p>Consider the Personalized Project Reference and Written Response 2(b) when scoring this point.</p> <ul style="list-style-type: none"> • If multiple procedures are included in part (i) of the Procedure section of the Personalized Project Reference, use the first procedure to determine whether the point is earned. <p>Do NOT award a point if any one or more of the following is true:</p> <ul style="list-style-type: none"> • 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 in part (i) of the Procedure section of the Personalized Project Reference. • The response describes expected behavior that is implausible, inaccurate, or inconsistent with the program.

Reporting Category	Scoring Criteria	Decision Rules
<p>Written Response 2(c): Data and Procedural Abstraction</p> <p>(0–1 points)</p>	<p>The written response:</p> <ul style="list-style-type: none"> explains in detailed steps an algorithm that uses <code>isEqual</code> to count the number of elements in the list that are equal to a certain value. 	<p>Consider the Personalized Project Reference and Written Response 2(c) when scoring this point.</p> <ul style="list-style-type: none"> If multiple lists are included in the List section of the Personalized Project Reference, use the first list to determine whether the point is earned. The algorithm can be described in code, pseudocode, as a sequence of steps in English, or as a paragraph in English. The algorithm must describe iterating over all elements of the list, calling <code>isEqual</code> on each list element and maintaining a count of the number of list elements equal to a certain value. The algorithm must correctly determine the number of elements that are equal to a certain value. Some ways this can be determined is by storing the value in a variable, returning it, or displaying it to the user. <p>Do NOT award a point if the following is true:</p> <ul style="list-style-type: none"> A list is not identified in the List section of the Personalized Project Reference. The description of the algorithm is not detailed enough for someone else to write the code.

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.

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.

Program 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.

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: Selection determines which parts of an algorithm are executed based on a condition being true or false. The use of try / exception statements is a form of selection 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”.