

**Video, Program Requirements, and Written Response 1****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
<b>Course Project: Video</b>  <b>(0–1 points)</b>	The video demonstrates the running of the program including: <ul style="list-style-type: none"> <li>• <i>input</i></li> <li>• <i>program functionality</i></li> <li>• <i>output</i></li> </ul>	<b>Consider only the video when scoring this point.</b>  <b>Do NOT award a point if the following is true:</b> <ul style="list-style-type: none"> <li>• The video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited).</li> </ul>
<b>Course Project: Program Requirements</b>  <b>(0–1 points)</b>	The program code includes: <ul style="list-style-type: none"> <li>• <i>A list</i></li> <li>• <i>A procedure</i></li> <li>• <i>A call to the procedure</i></li> <li>• <i>Selection</i></li> <li>• <i>Iteration</i></li> </ul>	<b>Consider the Personalized Project Reference (or Program Code if necessary) when scoring this point.</b> <ul style="list-style-type: none"> <li>• If the program requirements do not appear in the Personalized Project Reference, consider the full program code file when scoring this point.</li> <li>• The procedure does not need to have a parameter to earn this point.</li> <li>• The code segments demonstrating selection and iteration do not need to appear in the same algorithm to earn this point.</li> <li>• The code segments demonstrating selection and iteration do not need to be contained in a procedure to earn this point.</li> </ul> <b>Do NOT award a point if any one or more of the following is true:</b> <ul style="list-style-type: none"> <li>• The list is a one-element list.</li> <li>• The use of the list is irrelevant or not used in the program.</li> </ul> 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
<p><b>Written Response 1: Program Function and Purpose</b></p> <p><b>(0–1 points)</b></p>	<p>The written response:</p> <ul style="list-style-type: none"> <li>identifies an expected user of the program.</li> <li>describes one way the program’s design meets the needs of the identified user.</li> </ul>	<p><b>Consider Written Response 1 and the student’s Program Code when scoring this point.</b></p> <ul style="list-style-type: none"> <li>The response must relate the program design to the needs of the identified user.</li> </ul> <p><b>Do NOT award a point if the following is true:</b></p> <ul style="list-style-type: none"> <li>The description of the design is implausible, inaccurate, or inconsistent with the program.</li> </ul>

## 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”.