

---

# AP<sup>®</sup> Computer Science Principles

## Sample Student Responses and Scoring Commentary Set 2

### **Inside:**

#### **Written Response 1**

- ☒ **Scoring Guidelines**
- ☒ **Student Samples**
- ☒ **Scoring Commentary**

**Digital Portfolio Components provided separately**

**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
<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 the video (or Program Code if necessary) when scoring this point.</b> <ul style="list-style-type: none"> <li>• The video needs to show at least one example of the program’s functionality.</li> <li>• If the source of the input is unclear from the video, consider the full program code file when scoring this point.</li> </ul> <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 student-developed procedure</i></li> <li>• <i>A call to the student-developed procedure</i></li> <li>• <i>A list (or other collection type)</i></li> <li>• <i>A use of the list</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 <i>parameter</i> to earn this point.</li> <li>• The <i>code segments</i> demonstrating selection and iteration do not need to appear in the same <i>algorithm</i> 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 (i.e., not connected to the program’s functionality).</li> <li>• The call to the procedure is inconsistent with the procedure header (unless allowed by the programming language).</li> <li>• The use of either the selection or the iteration is trivial (i.e., does not affect the outcome of the program).</li> </ul>

Reporting Category	Scoring Criteria	Decision Rules
<b>Written Response 1: Program Design, Function, and Purpose</b>  <b>(0–1 points)</b>	<p>The written response:</p> <ul style="list-style-type: none"> <li>identifies an invalid or unexpected input to the program.</li> <li>describes the behavior of the program after it receives the identified input.</li> </ul> <p>OR</p> <ul style="list-style-type: none"> <li>explains why it is not possible for the program to accept invalid or unexpected input.</li> </ul>	<p><b>Consider the Video and Written Response 1 when scoring this point.</b></p> <ul style="list-style-type: none"> <li>If the video is not available, consider the full program code file when scoring this point.</li> <li>Either a specific example of the input or a description of the input can meet this criteria.</li> <li>If the response identifies more than one invalid or unexpected input, it only needs to correctly describe the behavior of the program for one of these inputs.</li> </ul> <p><b>Do NOT award a point if any one or more of the following is true:</b></p> <ul style="list-style-type: none"> <li>The identified invalid or unexpected input is implausible, inaccurate, or inconsistent with the program.</li> <li>The response does not provide details of the behavior of the program after it receives an invalid or unexpected input (e.g., by stating that it just doesn't work) or does not give a plausible explanation why the program cannot accept invalid or unexpected input.</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.

**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”.

An unexpected or invalid input that a user could provide to my program is entering "ten" for the input of the variable numitems (in the procedure "getprompt"). For example, if the user had chosen the words mode, they would be asked "Enter the number of words you would like in the prompt: ". Expected inputs are integers (example: 10), not strings of spelled-out numbers (example: ten), because the value of the input would later be used as the parameter in a for loop. In the case that the user enters an input other than an integer, such as "ten", the program would print "Please enter a valid number." The program behaves this way because the input statement is inside a while True loop meaning that the loop will keep iterating until it encounters the break statement. If and else statements are used to determine if the input is an integer (if numitems.isnumeric():) in which it will break the while True loop and move on, or if it is not an integer in which it will print "Please enter a valid number." and iterate the loop again.

The premise of the program is a flag guessing game, where the user is allowed to input any characters into the text box. It is not possible to accept an invalid input because each input is either incorrect or correct, however, there are some possible unexpected inputs. First, the program only supports English; if the user types answers in a different language, the user's answer will be compared to the established "correct" English answer and it will be marked as incorrect. Additionally, if the user does not capitalize the first letter of a technically correct answer (eg. "Canada" vs. "canada"), it will also be marked incorrect because I didn't add a line that turns the input into all lower / upper case letters before comparing it to the correct answer. If the input is different in any way to the string established, it will be marked as incorrect.

if you enter an input such as "Hieght" or "weight" with a incorrect charracter such as a letter or special charracter, the code does not accept it. It doesnt exepct the incroect character because the software in which the code is on will show the charracter as invalid upon attempting to complete the input value by clicking next. On other platforms or in general it may not allow the input if incorrect to go through, or if its inputed it will created an error when calculated against the other input making it imossible for a result aside from error to come out as the rsult.

It is not possible for my program to accept an unexpected or invalid input because when you try to take something off the list and it is not there, the program notices and put an alert on the screen if the input is not found on the list. This means that with any input, the program will always check if it is on the list and when the input is on the list, it will delete the input off the list.

In my program, An invalid input that an user could provide would be choosing the wrong answer. My program is a quiz app. therefore there are many options and some are incorrect. In the event that the user does choose something wrong. The app will not allow the user to move on. As said, The response will be that the screen will not turn. until, the right answer has been provided.

A user could input an x or y value into my program that is  $<0$ , if that is the case, it would cause the program to not run due to the fact that there is no negative positions within the canvas provided. In the event that an x or y value is a number  $>400$ , the code will continue to run, yet the heart would be displayed out of bounds for the canvas. While the heart would still exist, it would not be visible due to the limitations of the canvas view. If the user was to input a color not on the registry of colors through the CMU coding program, or a color with an grb code that doesn't exist, the color wouldn't fill and the code wouldn't work. If a color is chosen outside of the list: `app.color`, then the border of said heart will not correlate with the fill color.

## Question 1

**Note:** Student samples are quoted verbatim and may contain spelling and grammatical errors.

### Overview

**NEW for 2025:** The question overviews can be found in the *Chief Reader Report on Student Responses on AP Central*.

**Sample: A**

**Score:**

**Video: 1**

**Program Requirements: 1**

**Question 1: 1**

Video:

The response earned this point, demonstrating all three criteria:

- The response demonstrates input by showing the user typing characters.
- The response demonstrates program functionality by displaying text to test a user’s typing ability.
- The response demonstrates output by displaying the results of the typing test.

Program Requirements:

The response earned this point, demonstrating all six criteria:

- The response includes a student-developed procedure: `getprompt(mode)`
- The response calls the procedure: `getprompt(mode)`
- The response includes selection: `if numitems.isnumeric()`
- The response includes iteration: `for i in range(int(numitems))`
- The response includes a list: `words` (which is later assigned to the variable `modelist`)
- The response shows the list `modelist` being used in the Program Code.

Question 1:

The response earned this point, demonstrating both of the criteria:

- The response identifies an invalid input as “strings of spelled-out numbers (example: ten).”
- The response describes the behavior of the program after it receives the input: “If and else statements are used to determine if the input is an integer (`if numitems.isnumeric():`) in which it will break the while True loop and move on, or if it is not an integer in which it will print “Please enter a valid number.” and iterate the loop again.”

**Question 1 (continued)****Sample: B****Score:****Video: 1****Program Requirements: 1****Question 1: 1**

Video:

The response earned this point, demonstrating all three criteria:

- The response demonstrates input by typing a country in an input box.
- The response demonstrates program functionality displaying to the user when they correctly or incorrectly match a flag.
- The response demonstrates output by displaying the flag a user must match.

Program Requirements:

The response earned this point, demonstrating all six criteria:

- The response includes a student-developed procedure: `EndFunction(UserList)`
- The response calls the procedure: `EndFunction(Guesses)`
- The response includes selection: `if (UserList[i] == flagNames[i])`
- The response includes iteration: `for (var i=0; i < Guesses.length; i++)`
- The response includes a list or collection: `flagNames`
- The response shows the list being used in part ii of the List section of the PPR.

Question 1:

The response earned this point, demonstrating both of the criteria:

- The response identifies a couple unexpected inputs to the program, including: “if the user types answers in a different language, the user’s answer will be compared to the established “correct” English answers.”
- The response describes the behavior of the program after receiving the unexpected input: “answer will be marked as incorrect.”

**Question 1 (continued)****Sample: C****Score:****Video: 1****Program Requirements: 0****Question 1: 1**

Video:

The response earned this point, demonstrating all three criteria:

- The response demonstrates input by showing the user entering the text “196” in the text box and clicking OK.
- The response demonstrates program functionality by showing pop-up windows asking questions.
- The response demonstrates output by showing the BMI calculation.

Program Requirements:

The response did not earn this point, demonstrating three of the six criteria:

- The response includes a student-developed procedure: `getTheBMI()`
- The response calls the procedure: `getTheBMI()`
- The response includes selection: `if (isNaN(weight) || weight <= 0)`
- The response does not include iteration.
- The response does not include a list.
- The response does not show a list being used.

Question 1:

The response earned this point, demonstrating both criteria:

- The response identifies “a incorrect charracter such as a letter or special charracter” as invalid or unexpected input.
- The response describes the behavior after receiving invalid input by stating that the program “will show the charracter as invalid upon attempting to complete the input value by clicking next.”

**Question 1 (continued)****Sample: D****Score:****Video: 1****Program Requirements: 0****Question 1: 1**

Video:

The response earned this point, demonstrating all three criteria:

- The response demonstrates input by showing the user type into a text box.
- The response demonstrates program functionality by clicking the add button and displaying the item that had been typed on the “To Do list.”
- The response demonstrates output by displaying, “Do My math homework,” on the “To Do list.”

Program Requirements:

The response did not earn this point, demonstrating five out of six criteria:

- The response includes a student-developed procedure: `updateTaskList()`
- The response calls the procedure: `updateTaskList()`
- The response includes selection: `if (task != "")`
- The response does not include iteration.
- The response includes a list: `tasks`
- The response shows the list being used in part ii of the List section of the PPR.

Question 1:

The response earned this point.

The response states: “It is not possible for my program to accpet an unexpected or invalid input.” The response explains why: ”with any input, the program will always check if it is on the list and when the input is on the list, it will delete the input off the list.”

**Question 1 (continued)****Sample: E****Score:****Video: 1****Program Requirements: 0****Question 1: 0**

Video:

The response earned this point, demonstrating all three criteria:

- The response demonstrates input by the user clicking the buttons on the screen.
- The response demonstrates program functionality by displaying quiz questions and answer options.
- The response demonstrates output by displaying a secret message.

Program Requirements:

The response did not earn this point, demonstrating two of the six criteria:

- The response includes a student-developed procedure: `StartScreen()`
- The response calls the procedure: `StartScreen()`
- The response includes a selection statement: `If (0==0)` (though this is a trivial selection statement)
- The response does not include iteration.
- The response does not include a list or collection.
- The response does not show a list or collection being used in the PPR or Program Code.

Question 1:

The response did not earn this point, demonstrating one of the two of the criteria:

- The response does not identify an invalid or unexpected input to the program. The response states, “An invalid input that an user could provide would be choosing the wrong answer.” However, in a quiz app the user is expected to click buttons and enter incorrect answers, so this is an expected input to the program.
- The response describes the behavior of the program after it receives the identified input: “In the event that the user does choose something wrong. The app will not allow the user to move on.”

**Question 1 (continued)****Sample: F****Score:****Video: 0****Program Requirements: 1****Question 1: 0**

Video:

The response did not earn this point, demonstrating two of three criteria:

- The response does not demonstrate input.
- The response demonstrates program functionality by displaying the hearts on the screen and moving to different locations on the screen.
- The response demonstrates output by displaying hearts on the screen.

Program Requirements:

The response earned this point, demonstrating all six criteria:

- The response includes a student-developed procedure: `drawHearts(x, y, choice)`
- The response calls the procedure: `drawHearts(randrange(0, 400), randrange(0, 400), choice(app.color))`
- The response includes selection: `if (choice=='lightPink')`
- The response includes iteration: `for h in heart.children`
- The response includes a list or collection: `heart`
- The response shows the collection being used in the Program Code.

Question 1:

The response did not earn this point, demonstrating one of the two criteria:

- The response identifies user input for “an x or y value into my program that is <0.” However, there is no input in the program so this input is inconsistent with the program.
- The response describes how the invalid input would cause the heart to “not be visible due to the limitations of the canvas view.”