

AP CSP Test Breakdown

No comments in Personalized Project Reference

Section	Question Type / Component	Number of Questions	Exam Weighting	Timing
I	Multiple-choice questions	70	70%	120 minutes End-of-course AP Exam
	• Single-select	57		
	• Single-select with reading passage about a computing innovation	5		
	• Multi-select	8		
II	Create Performance Task	See Below	30%	See Below
	• Program code, video, and Personalized Project Reference			At least 9 hours in class
	• Written response related to the Create performance task	4		60 minutes End-of-course AP Exam

- An in-class through-course project that is **30% of the AP score**.
- A programming project where students will design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity.
- The programming language students use is their choice, but we recommend they use the language that is being used in class. This is not the time to learn another programming language.
- This is a project in which students can collaborate with another AP CSP student to complete the programming portion.
- Students will have **at least 9** classroom hours to complete this task.
- Scored on a **6-points scoring guideline**:
 - 2 points are related to the Program Code and Video.
 - 4 points are related to the four written response questions students will answer on exam day.

AP CSP End-of-Course Exam

What's Changing?

- Students will have 1 hour dedicated to completing the written response questions
- Each of the 4 questions will correspond to one of the following categories:
 - Program Design, Function, and Purpose
 - Algorithm Development
 - Errors and Testing
 - Data and Procedural Abstraction

What's Staying the Same?

- The multiple-choice exam section will still be 2 hours and 70 multiple-choice questions
- The weighing for the multiple-choice section will remain 70% while the Create PT will continue to be 30%
- The scoring guidelines will continue to have 6 equally weighted rubric rows

What's Changing?

- Reduction in the number of in-class hours required for the Create performance task from 12 hours to 9 hours
- The Written Response Portion of the Exam will now be included in the end-of-course exam
- The Written Response Questions will be unique on every form of the exam
- Scoring guidelines will not be available in advance
- Teachers can use practice questions with their students to practice the written response. Teachers can only give feedback after all components are submitted as final in the AP Digital Portfolio
- Students will create a Personalized Project Reference that will contain the required code related to a list and procedure. This will be captured in the digital portfolio. AP Coordinators will need to print this reference sheet and provide it to students on exam day

What's Staying the Same?

- Students will still generate a program of their choice that meets the current program requirements
- Students will still create a video of their program running
- Students will still submit their program code and video to the digital portfolio in advance of the exam

Components of the Create Performance Task

There are **three components** submitted to the Digital Portfolio:

- A** **Program Code** can be created collaboratively;
 - the program must meet the requirements outlined in the Student Handouts
- B** **Video**
 - created independently;
 - the video must meet the requirements outlined in the Student Handouts
- C** **Personalized Project Reference**
 - created independently;
 - contains only your procedure and list;
 - cannot contain any comments in the code;
 - will be supplied to you on exam day by your AP Coordinator

Component A: Program Code Requirements

In your program, you must include student-developed program code that contains the following:

- **Instructions for input from one of the following:**
 - the user (including user actions that trigger events)
 - a device
 - an online data stream
 - a file
- Use of at least one **list** (or other **collection type**) to represent a collection of data that is stored and used to manage program complexity and help fulfill the program's purpose
- At least one procedure that contributes to the program's intended purpose, where you have defined:
 - the procedure's name
 - the return type (if necessary)
 - one or more parameters
- An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- Calls to your student-developed procedure
- Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

Component B: Video Requirements

Your video must demonstrate your program running, including:

- Input to your program
- At least one aspect of the functionality of your program
- Output produced by your program

Your video may NOT contain:

- Any distinguishing information about yourself
- Voice narration (though text captions are okay)

Your video must be:

- Either .mp4, .wmv, .avi, or .mov format
- No more than 1 minute in length
- No more than 30MB in file size

Component C: Personalized Project Reference

Procedure

Capture and paste two program code segments you developed as part of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

i

The first program code segment must be a student-developed procedure that:

- Defines the procedure's name and return type (if necessary)
- Contains and uses one or more parameters that have an effect on the functionality of the procedure
- Implements an algorithm that includes sequencing, selection, and iteration

ii

The second program code segment must show where your student-developed procedure is being called in your program.

Component C: Personalized Project Reference

List

Capture and paste two program code segments you developed as part of this task that contain a list (or other collection type) being used to manage complexity in your program.

i

The first program code segment must show how data have been stored in the list.

ii

The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Plagiarism

The use of program code, media (e.g., video, images, sound), data, information, or evidence created by someone else or with generative AI tools in the creation of a program and/or a program code segment(s), without appropriate acknowledgment (i.e., through citation, through attribution, and/or by reference), is considered **plagiarism**.

A student who commits plagiarism will receive **a score of 0 on the Create performance task**, including their responses to the written response prompts on the end-of-course AP Exam.

Acceptable Generative AI Use

Students are permitted to utilize Generative AI tools as supplementary resources for understanding coding principles, assisting in code development, and debugging. This responsible use aligns with current guidelines for peer collaboration on developing code.

Students should be aware that Generative AI tools can produce:

- incomplete code;
- code that creates or introduces biases;
- code with errors;
- inefficiencies in how the code executes; or
- code complexities that make it difficult to understand and therefore explain the code.

It is the student's responsibility to review and understand any code co-written with AI tools, ensuring its functionality. Additionally, students must be prepared to explain their code in detail, as required on the end-of-course AP Exam.

Preparing for Final Submission

Students **are not permitted** to collaborate on the **video** or creation of the **Personalized Project Reference**.

The Personalized Project Reference should be limited to code that addresses the requirements of the task. The Personalized Project Reference cannot include comments within the code or on any other part of the reference.

Including comments in the Personalized Project Reference will result in students receiving a **score of 0** on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

Attestations

During the final submission process in the AP Digital Portfolio, **students will be asked to attest** that they have followed the Performance Task guidelines and have not plagiarized their submission.

Each of the **three components** of the Create performance task must be **submitted as final** to be sent for scoring.

Failing to submit the Personalize Project Reference as final in the AP Digital Portfolio by the deadline will mean you will not have access to this resource on exam day.

Before beginning the Create Performance Task, be sure to:

- **Review** the performance task directions and guidelines.
- **Brainstorm problems** that programming can address, or brainstorm special interests that programming can help develop.
- As needed, **seek assistance from your teacher or AP Coordinator** on defining your focus and choice of topics.
- Be prepared to **collaborate** with peers as necessary.
- **Practice and discuss** all or part of the performance task, including practice written response prompts.
- Take advantage of the opportunity to **get assistance and feedback** from your teacher during practice. Your teacher cannot provide any feedback or assistance on your Create performance task components or sample writing until after everything is submitted as final in the AP Digital Portfolio.
- Ensure you know the proper way to **cite program code or media**, including APIs or other pieces of open-source code, used in the creation of your program. Any media or program code that has not been written by you must be cited, and credit must be given to the author. Any existing program code needs to be extended in some new way by adding new functionality.
- Read through the AP Digital Portfolio file submission requirements and process.
- Practice making videos and readable screen captures.

Guidelines for Completing the Create PT

You MUST:

- Submit all three components of your performance task as final prior to the submission deadline, which can be found on the AP Computer Science Principles Exam page on AP Central. Only files submitted as final will be sent for scoring.
- Follow a calendar or schedule that provides time for all performance task components to be completed and uploaded in advance of the deadline.
- Read the performance task directions.
- Apply the computer science knowledge you have obtained throughout the course, and when completing the performance task.
- Use acceptable acknowledgment practices when using program code, media (i.e., images, videos, sound), or data sources created by others or with generative AI tools in your program code to avoid plagiarism. Program code not written by you could include starter code provided by your teacher, the use of APIs or open-source code, or generated using an AI tool. When using existing code, you must extend the program in some new way by adding new functionality. Some examples of ways to provide attribution for program code that was not authored by you are as follows:
 - If the program code has been made available for your use by your teacher, add a comment that states: *This code was provided as starter code by my teacher.*
 - If the program code has been made available through an API or open-source code, add a comment that states: *This code was made freely available by [source of code].*
 - If the program code has been co-created with the assistance of a generative AI tool, add a comment that states: *This code was generated using [Generative AI Tool Name].*
- Add comments to acknowledge and credit authors of media, data sources, or program code:
 - If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author.
 - If the programming environment does not allow you to include comments, you can add them in a document editor when you capture your program code for submission.

Guidelines for Completing the Create PT

You MAY NOT:

- Seek assistance in writing, revising, amending, or correcting your work, including debugging or error testing the program, writing or designing functionality in the program, testing the program, or making revisions to the program, from anyone other than your collaborative partner(s).
- Submit practice performance tasks or any work that has been revised, amended, or corrected by another individual, other than your collaborative partner(s) or cited program code, as a submission for AP Exam scoring.
- Include any course content in the screen captures of the program code included in the Personalized Project Reference. Any comments that were included in the program code during the development of your program should be removed prior to taking screen captures. Including comments in your screen captures will result in the written response portion of your exam being scored a 0.
- Collaborate during the creation of your video or creation of your Personalized Project Reference.
- Revise your work once you have completed and submitted it as final to the AP Digital Portfolio.

Guidelines for Completing the Create PT

You MAY:

- Collaborate with your partner(s) only during the ideation and development, including debugging and error testing, of your program code, if you choose to do so. **NOTE: You are not allowed to collaborate on your video or creation of your Personalized Project Reference.**
- Seek assistance from your teacher or AP Coordinator on the **formation of groups and resolution of collaboration issues** when one collaborative partner is clearly and directly impeding the completion of the performance task.
- Seek clarification from your teacher or AP Coordinator on the **program requirements and submission requirements** for the performance task when you do not understand the directions.
- Work on the performance task **outside of designated class time**.
- Seek assistance from your teacher or AP Coordinator to **resolve technical problems that impede work**, such as a failing workstation or difficulty with access to networks, or to help with saving files or making movie files.
- **Keep a programming journal** of the design choices that were made during the development of the program code or code segment and the effect of these decisions on the program's function.

Legible Submissions

- Be sure that the code in your Program Code is legible.
 - Use at least 10-point font.
- Be sure that the Video is not blurry.
 - Ensure that the screen size isn't too small when you are recording
 - Using screen capture video software will help to ensure that the video is not shaky.
 - Do not attempt to speed up your video to fit within the 1-minute requirement as this may make it difficult to view
 - Ensure your video is a good quality
- Be sure that the code in your Personalized Project Reference is legible.
 - **Remember you will need to read from a printed copy of your Personalized Project Reference on exam day.**
 - Use at least 10-point font.

Attestations for Final Submission

Submit all three components as final in the AP Digital Portfolio

Students will need to attest:

- That the work submitted is their own original work and any work developed with peers or generative AI tools is properly attributed, and that they have read and understood the AP CSP policy on plagiarism.
- That they have read the [AP Computer Science Principles Student Handouts](#) (.pdf/1.06 MB) document.
- That they did not collaborate on the individual video or the creation of Personalized Project Reference for the performance task.
- Their Personalized Project Reference does not contain comments. It only includes program code used in the student's program.

Task Verbs

The following task verbs are commonly used in the performance task and written responses:

- **Capture:** Select a portion of program code that addresses the prompt(s).
- **Demonstrate:** Provide sufficient evidence for an answer or point being made.
- **Describe:** Provide the relevant features or characteristics of what the program code represents or is being used to accomplish.
- **Design:** Develop a plan for how to accomplish the program specification or requirements.
- **Explain:** Provide information about how or why a relationship, situation, or outcome occurs, listing detailed steps of the algorithm or using evidence and/ or reasoning.
- **Identify:** Provide a name for the specific topic, without elaboration or explanation.
- **Implement/Write:** Recognize and use proper syntax to execute the program design.

Categories for The Written Response

Program Design, Function, and Purpose

Students should be prepared to respond to prompts about their program that assess any of the following learning objectives:

- **CRD-2.A:** Describe the purpose of the computing innovation.
- **CRD-2.B:** Explain how a program or code segment functions.
- **CRD-2.C:** Identify input(s) to a program.
- **CRD-2.D:** Identify output(s) produced by a program.
- **CRD-2.E:** Develop a program using a development process.
- **CRD-2.F:** Design a program and its user interface.
- **CRD-2.G:** Describe the purpose of a code segment or program by writing documentation.

Program Design, Function, and Purpose

An example question from this category is:

Identify an expected user of your program. Describe one way your program's design meets the needs of this user.

Categories for The Written Response

Algorithm Development

Students should be prepared to respond to prompts about their program that assess any of the following learning objectives:

- **CRD-2.B:** Explain how a program or code segment functions.
- **AAP-2.E.b:** Evaluate expressions that use relational operators.
- **AAP-2.F.b:** Evaluate expressions that use logic operators.
- **AAP-2.H.b:** Determine the result of conditional statements.
- **AAP-2.J:** Express an algorithm that uses iteration without using a programming language.
- **AAP-2.K.b:** Determine the result or side effect of iteration statements.
- **AAP-2.L:** Compare multiple algorithms to determine if they yield the same side effect or result.
- **AAP-2.M.a:** Create algorithms.
- **AAP-2.M.b:** Combine and modify existing algorithms.

Algorithm Development

An example question from this category is:

Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Identify the number of times the body of your iteration statement will execute. Describe a condition or error that would cause your iteration statement to not terminate and cause an infinite loop. If no such condition or error exists, explain how the loop could be modified to cause an infinite loop.

Mostly loop, not recursion.

Categories for The Written Response

Errors and Testing

Students should be prepared to respond to prompts about their program that assess any of the following learning objectives:

- **CRD-2.I.a:** Identify the error.
- **CRD-2.I.b:** Correct the error.
- **CRD-2.J:** Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.

Errors and Testing

An example question from this category is:

Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Describe a change to your procedure that will result in a run-time error. Explain why this change will result in a run-time error.

Categories for The Written Response

Data and Procedural Abstraction

Students should be prepared to respond to prompts about their program that assess any of the following learning objectives:

- **AAP-1.D.a:** Develop data abstraction using lists to store multiple elements.
- **AAP-1.D.b:** Explain how the use of data abstraction manages complexity in the program.
- **AAP-2.O.a:** Write iteration statements to traverse a list.
- **AAP-2.O.b:** Determine the result of an algorithm that includes list traversals.
- **AAP-3.B:** Explain how the use of procedural abstraction manages complexity in a program.

Data and Procedural Abstraction

An example question from this category is:

Suppose you are provided with a procedure called `isEqual(value1, value2)`. The procedure returns `true` if the two parameters `value1` and `value2` are equal in value and returns `false` otherwise. Using the list you identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `isEqual` to count the number of times a certain value appears in your list. Your explanation must be detailed enough for someone else to write the program code.

Course Project: Video

Course Project: Video (0 - 1 points)	The video demonstrates the running of the program including: <ul style="list-style-type: none">• <i>input</i>• <i>program functionality</i>• <i>output</i>	Consider only the video when scoring this point. Do NOT award a point if the following is true: <ul style="list-style-type: none">• 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

Course Project: Program Requirements (0 - 1 points)	The program code includes: <ul style="list-style-type: none">• A <i>list</i>• A <i>procedure</i>• A <i>call to the procedure</i>• <i>Selection</i>• <i>Iteration</i>	Consider the Personalized Project Reference (or Program Code if necessary) when scoring this point. <ul style="list-style-type: none">• 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 parameter to earn this point.• The code segments demonstrating selection and iteration do not need to appear in the same algorithm 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: <ul style="list-style-type: none">• The list is a one-element list.• The use of the list is irrelevant or not used in the program• The use of either the selection or the iteration is trivial (i.e., does not affect the outcome of the program).
----------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Written Response 1: Program Function and Purpose

Prompt:

Identify an expected user of your program.
Describe one way your program's design meets the needs of this user.

Written Response 1: Program Function and Purpose (0 - 1 points)	The written response: <ul style="list-style-type: none">• identifies an expected user of the program• describes one way the program's design meets the needs of the identified user.	<p>Consider Written Response 1 and the student's Program Code when scoring this point.</p> <ul style="list-style-type: none">• The response must relate the program design to the needs of the identified user. <p>Do NOT award a point if the following is true:</p> <ul style="list-style-type: none">• The description of the design is implausible, inaccurate, or inconsistent with the program.
--------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Written Response 1: Program Function and Purpose

Sample Response

The user of my game would be children who are trying to learn their colors in another language.

The program design is great for children because each color button is filled with the corresponding color for the word. This will make it easy for students who don't know how to read to still use the app to hear the colors in languages.

Written Response 2(a): Algorithm Development

Prompt:

Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Identify the number of times the body of your iteration statement will execute. Describe a condition or error that would cause your iteration statement to not terminate and cause an infinite loop. If no such condition or error exists, explain how the loop could be modified to cause an infinite loop.

Written Response 2(a): Algorithm Development (0 - 1 points)	<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.
------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Written Response 2(a): Algorithm Development

Example 1:

Consider the following sample code segment used for illustrative purposes only:

```
for (int i = 1; i <= 5; i++)  
{   System.out.println (i); }
```



Sample Response

The loop iterates five times.

And infinite loop would occur if the loop condition `i <= 5` was changed to `i > 0`, since `i` starts at 1 and is always incremented by 1.

Written Response 2(a): Algorithm Development

Example 2:

Consider the following sample code segment used for illustrative purposes only:

```
for (int i : listOfValues)
{   System.out.println (i); }
```

Sample Response

The loop iterates for each element in `listOfValues`.

An infinite loop is not possible since this for each loop iterates over all the elements of the list. If the list does not contain any elements, it would iterate 0 times.

Written Response 2(b): Errors and Testing

Prompt:

Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Describe a change to your procedure that will result in a run-time error. Explain why this change will result in a run-time error.

<p>Written Response 2(b): Errors and Testing (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 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.
-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Written Response 2(b): Errors and Testing

Consider the following sample code segment used for illustrative purposes only:

```
Procedure printNums (int stop) {  
    for (int i = 1; i <= stop; i++)  
    { System.out.println (i); }  
}
```

Sample Response

A run-time error would occur if instead of printing `i`, we printed `i / 0`, since this would cause a divide by 0 error.

Written Response 2(c): Data and Procedural Abstraction

Prompt:

Suppose you are provided with a procedure called `isEqual(value1, value2)`. The procedure returns `true` if the two parameters `value1` and `value2` are equal in value and returns `false` otherwise. Using the list you identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `isEqual` to count the number of times a certain value appears in your list. Your explanation must be detailed enough for someone else to write the program code.

Written Response 2(c): Data and Procedural Abstraction (0 - 1 points)	The written response: <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.	Consider the Personalized Project Reference and Written Response 2(c) when scoring this point. <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 the 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. Do NOT award a point if the following is true: <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.
------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Written Response 2(c): Data and Procedural Abstraction

Sample Response

Set a counter variable count to 0.
Iterate over the elements of the list named myList.
For each element, call isEqual to compare the element from myList to the given value. If isEqual returns true add 1 to count.
After iterating over every list value, print the value of count.

Tips for Teachers and Students

- Assign at least one sample set of written response prompts prior to students submitting as final.
- **Teachers are not allowed to provide any feedback on the written response prompts or any components of the Create performance task until after all three components are submitted as final to the AP Digital Portfolio.**
- **Prior to AP Exam day**, students should review their program code and video so they will remember the program's purpose, function, input and output. This is important in answering Written Response 1.
- Be sure to submit **all three components as Final** in the AP Digital Portfolio. Students who do not submit their Personalized Project Reference as final will not have access to this document on exam day.
- Be sure that **all course content and comments have been removed** from the screen captures used in the Personalized Project Reference. Teachers should return the files to students if they see course content or comments in the Personalized Project Reference. **If there is course content or comments in the Personalized Project Reference, your Create PT will be scored a 0.**

How can students cite code that came from somewhere else, if they need to remove all comments?

- Comments should be included in the program code.
- All code captured in the Personalized Project Reference video must be written by the student.