**Create Performance Task Written Response Prompts**

This document compiles the sample of prompts supplied by CollegeBoard as examples of the types of questions students may be asked to respond to regarding the program they created, using the information they captured on their Personal Project Reference (PPR).

**Writing Prompt #1: 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 a 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.

Question bank for Writing Prompt 1. This question accompanies the video and PDF of the complete code.

|  | Explain how the input to your program is received and how expected corresponding output(s) are produced as a result. |
| --- | --- |
|  | Describe the problem that your program was created to address or the creative expression it pursues. |
|  | Explain how you used or could have used feedback, testing or reflection in the development of your program |
|  | Describe one piece of documentation that would be appropriate to include with or in your program. Describe the intended purpose of this documentation by identifying who would use it and what they would do with it. |
|  | Identify an expected user of your program. Describe one way your program’s design meets the needs of this user. |
|  | Programs accept input to achieve their intended functionality. Describe at least one valid input to your program and what your program does with that input. |
|  | Identify the expected users of the program. Explain how your program addresses at least one concern or interest of the users you identified. |

**Writing Prompt #2a: 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.

Question bank for Writing Prompt 2a.

|  | Consider the first iteration statement included in the Procedure section of your PPR. Describe the condition(s) that will cause the body of the iteration statement to execute at least once. |
| --- | --- |
|  | Consider the first conditional statement included in the Procedure section of your PPR. Write an equivalent Boolean expression for this conditional statement. |
|  | Consider the first iteration statement included in the Procedure section of your PPR. Describe what is being accomplished by the code in the body of the iteration statement. |
|  | Consider the first conditional statement included in the Procedure section of your PPR. Describe your conditional statement, including its Boolean expression. Describe what the procedure does in general when the Boolean expression of this conditional statement evaluates to False. |
|  | Consider the code segment in part (ii) of the List section of your PPR that shows how your list is being used. Explain in detailed steps how this code segment works. Your explanation must be detailed enough for someone else to write the code segment. |
|  | Consider the first conditional statement included in the Procedure section of the PPR. Describe a condition that will cause the body of this conditional statement to execute and describe the result of the statement’s execution in the context of the program. |
|  | Consider the first iteration statement included in the Procedure section of your PPR. 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.  |

**Writing Prompt #2b: 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.

*NOTE: A runtime error happens when the program is running. A compile error or syntax error keeps the program from running.*

Question bank for Writing Prompt 2b.

|  | Consider the procedure included in part(i) of the Procedure section of your PPR. 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. |
| --- | --- |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Describe a call to the procedure and explain what behavior this call can be used to test in your program. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Write a call to your procedure with specific argument(s) that you could use for testing this procedure. Describe the program functionality that is related to this call. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Identify a strategy, other than using test cases, that you can use to test the correctness of your procedure. Describe how you would use this strategy. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Passing different values as arguments to a procedure can cause different segments of code to execute. Based on one of your arguments, describe where you could insert output statements in your procedure to test whether a block of code is executed or not. |
|  | Consider the procedure and procedure call identified in parts (i) and (ii) of the Procedure section of your PPR. Describe the outcome that your procedure call is intended to produce. Write a new procedure call with at least one different argument value that will produce the same outcome, if possible, and explain why this procedure call produces the same outcome. If it is not possible to write a new procedure call that produces the same outcome, explain why it is not possible. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Write two calls to your procedure that each cause a different code segment in the procedure to execute. Describe the expected behavior of each call. If it is not possible for two calls to your procedure to cause different code segments to execute, explain why this is the case for our procedure. |

**Writing Prompt #2c: Data and Procedural Abstraction**

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.

*NOTE: Data abstraction = lists; procedural abstraction = functions with parameters*

Question bank for Writing Prompt 2c.

|  | Suppose another programmer provides you with a procedure called **checkValidity(value)** that returns **true** if a value passed as an argument is considered valid by the other programmer and returns **false** otherwise. Using the list identified in the List section of your PPR, explain in detailed steps an algorithm that uses **checkValidity** to check whether all elements in your list are considered valid by the other programmer. Your explanation must be detailed enough for someone else to write the program code for the algorithm that uses **checkValidity**. |
| --- | --- |
|  | Consider the procedure identified in part(i) of the Procedure section of your PPR. Identify the parameter(s) used in this procedure. Explain how your identified parameter(s) use abstraction to manage complexity in your program. |
|  | 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 PPR, 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.  |
|  | Consider the list identified in the List section of your PPR. Describe how your program could be written without the use of a list. If it is not possible for your program to work without the use of a list, describe how your program’s functionality would be limited. |
|  | Consider the list identified in the List section of your PPR. Explain how you would need to adjust this part of your program if the list was not included in your code. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Explain how your program could be written differently if one of your parameters was removed from your procedure. |
|  | Consider the procedure identified in part (i) of the Procedure section of your PPR. Procedures are often used to organize larger problems into subproblems or smaller tasks. Identify the subproblem being solved or task that is being accomplished by your procedure. Explain how the procedure is used to accomplish the overall functionality of your program. |

.