**Create Performance Task**

**2020-2021**

**Overview**

Programming is a collaborative and creative process that brings ideas to life through the development of software. In this performance task, you will design and implement a program to solve a problem, enable innovation, explore personal interest, or express creativity. Your development process should include exploration, investigation, reflection, design, implementation, and testing your program.

You are strongly encouraged to work with another student in your class on the development of the program, only. However, the written response and the video that you submit for this performance task must be completed individually, without any collaboration with your partner or anyone else. Code provided in the written response parts 3b and 3c needs to be student-developed (can be collaboratively or individually developed) during the administration of the performance task.

*Please note that once this performance task has been assigned as an assessment for submission to College Board, you are expected to complete the task with minimal assistance from anyone with the exception of your collaborative peer(s) and then only when developing the program code. For more clarification see the Guidelines for Completing the Create Through-Course Performance Tasks section of the Course and Exam Description.*

**General Requirements**

You will be provided with a minimum of 12 hours of class time to complete and submit the following:

* your complete program code;
* a video (created independently) that displays the running of your program and demonstrates functionality you developed; and
* your individual written responses to all the prompts in the performance task.

Scoring guidelines and instructions for submitting your performance tasks are available on the AP Computer Science Principles Course Home Page.

Note: Students in non-traditional classroom environments should consult a school-based AP Coordinator for instructions.

**Submission Requirements**

**1.** **Program Code**

Your program must demonstrate:

* output (tactile, audible, visual, or textual) based on input from:
	+ the user (including user actions that trigger events); or
	+ a device; or
	+ a file;
* use of at least one list[[1]](#footnote-1) (or other collection type[[2]](#footnote-2))to represent a collection of data related to the program’s purpose; and
* development of at least one procedure that uses one or more parameters to accomplish the program’s intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

Create a PDF file that contains all of your program code (including comments).

**2. Video**

Your video must demonstrate your program running, including:

* input to your program; and
* at least one aspect of the functionality of your program; and
* output produced by your program.

Your video:

* must be either .mp4, .wmv, .avi, or .mov format; and
* must not exceed 1 minute in length; and
* must not exceed 30MB in file size.

Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

**3. Written Responses**

Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a–3d in order. Your response to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

3a. Provide a written response that:

* describes the overall purpose of the program; and
* describes what functionality the video illustrates; and
* describes the input and output shown in the video.

 (Approx. 150 words)

3b. Capture and paste two program code segments you developed during the administration of this task which contain a list[[3]](#footnote-3) (or other collection type[[4]](#footnote-4)) being used in your program. The first program code segment must show how data has been stored in the list1. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

* identifies the name of the list being processed in this response; and
* identifies what the data contained in the list is representing in your program; and
* explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

(Approx. xxx words)

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

* contain and use one or more parameters that have an effect on the functionality of the procedure; and
* implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

* describes what the selected procedure does and how it contributes to the overall functionality of the program; and
* explains how the algorithm implemented in the selected procedure accomplishes its task.

 (Approx. xxx words)

3d. Provide a written response that:

* describe two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
* describes what condition(s) is being tested by each call to the procedure; and
* identifies the result of each call.

(Approx. xxx words)

1. 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 names, such as arrays, depending on the programming language. [↑](#footnote-ref-1)
2. A collection type is a type that 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. [↑](#footnote-ref-2)
3. 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 names, such as arrays, depending on the programming language. [↑](#footnote-ref-3)
4. A collection type is a type that 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 [↑](#footnote-ref-4)