

Create Performance Task




Programming is a collaborative and creative process that brings ideas to life through the development of software. In the Create performance task, you will design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity. Your submission must include the elements listed in the Submission Requirements section below.

You are allowed to collaborate with your partner(s) on the development of the program only. **The video and Personalized Project Reference that you submit for this performance task must be completed individually, without any collaboration with your partner(s) or anyone else.** You can develop the code segments used in your **Personalized Project Reference** with your partner(s) or on your own as you work on the performance task during class.

Please note that once your teacher has assigned this performance task as one of your AP score components, you are expected to complete the task without assistance from anyone except for your partner(s) and then only when developing the program code. You must follow the Guidelines for Completing the Create Performance Task section below.

General Requirements

You will be provided with a minimum of 9 hours of class time to complete and submit the following:

-  ■ **Final program code** (created independently or collaboratively)
-  ■ **A video that displays the running of your program and demonstrates functionality you developed** (created independently)
-  ■ **Code Segments for your Personalized Project Reference** (created independently)

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

Submission Requirements



COMPONENT A: PROGRAM CODE (CREATED INDEPENDENTLY OR COLLABORATIVELY)

Submit one PDF file that contains all of your program code (including comments). 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).

IMPORTANT:

If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author. However, 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.

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

IMPORTANT:

The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

- ☐ 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

IMPORTANT:

Implementation of built-in or existing procedures or language structures, such as event handlers or main methods, are not considered student-developed.

- ☐ 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

DEFINITION:

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 may be referred to by different names, such as **arrays**, depending on the programming language.

DEFINITION:

Collection Type

A **collection type** is a type that aggregates elements in a single structure. Some examples include lists, databases, and sets.

IMPORTANT:

With text-based program code, you can use the print command to save your program code as a PDF file, or you can copy and paste your code to a text document and then convert it into a PDF file. With block-based program code, you can create screen captures that include only your program code, paste these images into a document, and then convert that document to a PDF. Screen captures should not be blurry, and text should be at least 10 pt font size.



COMPONENT B: VIDEO (CREATED INDEPENDENTLY) Submit one video file that demonstrates the running of your program as described below. Collaboration is not allowed during the development of your video.

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 allowed)

Your video must be:

- ☐ Either .webm, .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 (CREATED

INDEPENDENTLY) To assist in responding to the written response prompts on exam day, submit required portions of your code by capturing and pasting program code segments you developed during the administration of this task. Screen captures should not be blurry, and text should be at least 10-point font size. Your code segments should not include any comments. These code segments will be made available to you on exam day only if this component is submitted as final in the AP Digital Portfolio by the deadline.

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

IMPORTANT:

Built-in or existing procedures and language structures, such as event handlers and main methods, are not considered student-developed.

- 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.

List: Capture and paste two program code segments you developed during the administration 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.

DEFINITION:

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 may be referred to by different names, such as **arrays**, depending on the programming language.

DEFINITION:

Collection Type

A **collection type** is a type that aggregates elements in a single structure. Some examples include lists, databases, and sets.

IMPORTANT:

The data abstraction manages complexity by making the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).