

Performance Task: Create – Applications from Ideas

Overview

Programming is a collaborative and creative process that brings ideas to life through the development of software. Programs can help solve problems, enable innovations, or express personal interests. In this performance task, you will be developing a program of your choice. Your development process should include iteratively designing, implementing, and testing your program. You are strongly encouraged to work with another student in your class.

Please note that once this performance task has been assigned as an assessment (rather than as practice), you are expected to complete the task with minimal assistance from anyone other than your collaborative peer(s). For more clarification see the Guidelines for Completing the Through-Course Performance Tasks section.

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

- ▶ **A video of your program running**
- ▶ **Individual written responses about your program and development process**
- ▶ **Program code**

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

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

General Requirements

This performance task requires you to develop a program on a topic that interests you or one that solves a problem. During the completion of this performance task, you will iteratively design, implement, and test your program. You will provide written responses to prompts about your program and specific program code that are significant to the functionality of your program. It is strongly recommended that a portion of the program involve some form of collaboration with another student in your class, for example, in the planning, designing, or testing (debugging) part of the development process. Your program development must also involve a significant amount of independent work writing your program code, in particular, algorithm(s) and abstraction(s) that you select to use as part of your written response to describe how the program code segments help your program run.

You are required to:

- ▶ independently develop an algorithm that integrates two or more algorithms and that is fundamental for your program to achieve its intended purpose;
- ▶ develop an abstraction that manages the complexity of your program;
- ▶ create a video that displays the running of your program and demonstrates its functionality;
- ▶ write responses to all the prompts in the performance task; and
- ▶ submit your entire program code.

Program Requirements

Your program must demonstrate a variety of capabilities and implement several different language features that, when combined, produce a result that cannot be easily accomplished without computing tools and techniques. Your program should draw upon mathematical and logical concepts, such as use of numbers, variables, mathematical expressions with arithmetic operators, logical and Boolean operators and expressions, decision statements, iteration, and/or collections.

Your program must demonstrate:

- ▶ use of several effectively integrated mathematical and logical concepts, from the language you are using;
- ▶ implementation of an algorithm that integrates two or more algorithms and integrates mathematical and/or logical concepts; and
- ▶ development and use of abstractions to manage the complexity of your program (e.g., procedures, abstractions provided by the programming language, APIs).

Submission Requirements

1. Video

Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. **Your video must not exceed 1 minute in length and must not exceed 30MB in size.**

2. Written Responses

Submit one PDF file in which you respond directly to each prompt. **Clearly label your responses 2a–2d in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.**

Program Purpose and Development

2a. Provide a written response or audio narration in your video that:

- ◆ identifies the programming language;
- ◆ identifies the purpose of your program; and
- ◆ explains what the video illustrates.

(Must not exceed 150 words)

2b. Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)*

2c. Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3** below) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)*

2d. Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3** below). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*

3. Program Code

Capture and paste your entire program code in this section.

- › Mark with an **oval** the segment of program code that implements the algorithm you created for your program that integrates other algorithms and integrates mathematical and/or logical concepts.
- › Mark with a **rectangle** the segment of program code that represents an abstraction you developed.
- › Include comments or acknowledgments for program code that has been written by someone else.

Preparing for the Create Performance Task

Prior to your teacher administering this task, you should:

- ▶ Brainstorm problems that programming can address, or brainstorm special interests that programming can help develop;
- ▶ Ensure you understand the iterative nature of developing a computer program;
- ▶ Be prepared to collaborate with peers as necessary and in different ways;
- ▶ Ensure you are able to analyze program code and code segments and explain the function as it relates to the overall program;
- ▶ Know how to keep a programming journal of the design choices that you will make during the development of your program code and the effect of these decisions on the program's function. You can use this journal as a point of reference when demonstrating your understanding of how:
 - › an algorithm was built as part of the integration of two or more algorithms;
 - › a program functions differently with the inclusion of algorithms and abstractions;
 - › the inclusion of an abstraction has made their program code more compact, readable and/or reusable and how the program would operate differently without the inclusion of the abstraction;
- ▶ obtain programming support as necessary while practicing the skills needed to complete the performance task.

Guidelines for Completing the Create Performance Task

You must:

- ▶ be aware of the performance task directions, timeline, and scoring criteria;
- ▶ apply computer science knowledge you've obtained throughout the course when developing your program and in your explanation of its function;
- ▶ provide acknowledgment for program code used in your program that is not your own; and
- ▶ allow your own interests to drive your choice of program.

You may:

- ▶ follow a timeline and schedule for completing the performance task;
- ▶ seek clarification from your teacher or AP Coordinator pertaining to the task;
- ▶ seek clarification from your teacher or AP Coordinator regarding submission requirements;

- ▶ as needed, seek assistance from your teacher or AP Coordinator in defining your focus or choice of topics;
- ▶ 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 help with saving files or making movie files;
- ▶ obtain assistance from your teacher or AP Coordinator with the formation of peer-to-peer collaboration when completing the Create performance task;
- ▶ seek assistance from your teacher or AP Coordinator in resolving collaboration issues where one partner is clearly and directly impeding the completion of the Create performance task; and
- ▶ seek guidance from your teacher or AP Coordinator to use and acknowledge APIs or other pieces of open-source code. Program code not written by you can be used in programs as long as you are extending the project in some new way. You should provide acknowledgment and credit from program code you did not write.

You may not:

- ▶ collaborate on the video or any of the written responses;
- ▶ submit work that has been revised, amended, or correct by another individual, with the exception of cited program code;
- ▶ submit work from a practice performance task as your official submission to the College Board to be scored by the AP Program; or
- ▶ seek assistance or feedback on answers to prompts.