

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 partner. For more clarification see the Guidelines for Completing the Through-Course Assessment section.

You will be provided with 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 rubrics 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. It is strongly recommended that a portion of the program involve some form of collaboration with another student in your class. Your program development must also involve a significant amount of independent work in the planning and designing parts of the process.

You are required to:

- ▶ iteratively design, implement, and test your program;
- ▶ independently create at least one significant part of your program;
- ▶ create a video that displays the running of your program and demonstrates its functionality;

- ▶ write responses to questions about your program; and
- ▶ include 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 other 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.

(Approximately 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. (*Approximately 200 words*)

- 2c. Capture and paste the program code segment that implements an algorithm (marked with an **oval** in **section 3** below) that is fundamental for your program to achieve its intended purpose. Your code segment must include an algorithm that integrates other algorithms and integrates 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. (*Approximately 200 words*)
- 2d. Capture and paste the program code segment that contains an abstraction you developed (marked with a **rectangle** in **section 3** below). Your abstraction should integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. (*Approximately 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 citations for program code that has been written by someone else.