

AP Computer Science Principles: Create Task Requirements

Below are the requirements for the Create Performance Task in AP Computer Science Principles. These guidelines will help ensure your app meets all College Board criteria.

Create Task Requirements

Program Purpose and Function

Define the purpose of the app. Clearly explain what the app does and how it achieves its goal. Include a description of the intended audience.

Program Design and Development

Plan and document the app's development process. Include descriptions or diagrams showing how the app will work (flowcharts, pseudocode, etc.).

Program Code

Write and submit at least one algorithm in the app that includes sequencing, selection, and iteration. Explain the algorithm's purpose and how it functions in the app.

Abstraction

Create and utilize at least one abstraction in the app. This could be a function, procedure, or reusable code block that helps manage complexity and increases program functionality. Explain how this abstraction was developed and why it's important to the program.

Testing

Show evidence of testing your program to ensure it functions correctly. Include examples of any bugs or challenges encountered and how they were resolved.

Video Demonstration

Submit a 1-2 minute video demonstrating the app's functionality. The video should:

- Show the program running.
- Highlight at least one key feature of the app in action.

Written Responses

Answer the written prompts provided by the College Board, which include:

- Describing the program's purpose, functionality, and process.
- Explaining the algorithm and abstraction used.
- Reflecting on the development process and identifying how the app meets the task requirements.

Scoring Guidelines Overview

Students are assessed on:

1. Program Purpose and Function (Program meets stated goals, audience focus).
2. Data and Abstractions (Effective use of variables, lists, or reusable code).

3. Algorithm Implementation (Includes sequencing, selection, and iteration).

4. Program Testing and Debugging (Demonstrates problem-solving and refinement).

Tips for Students

- Choose a real-world problem or relatable idea to build your app around.
- Incorporate user interactivity (e.g., buttons, input fields, or dynamic data).
- Test frequently during development to catch and resolve errors early.
- Use comments in the code to clarify algorithms and abstractions for reviewers.