



AP Computer Science Principles - Create Task

Build something cute and clever

Kai kai@42.us.org

Summary: This project is about understanding the College Board grading scale for your Create Task project.

Contents

I	Introduction	2
II	Study the Task Requirements	3
II.1	Form a pair	3
II.2	Understand what you will turn in	3
II.3	Read the Manual	3
II.4	Grade practice examples	3
II.5	Check your grading against the key	4
III	Brainstorm your own project	5
IV	Coding	6
V	Resources for Finishing the Project	7

Chapter I

Introduction

One of the most important (and fun!) parts of the AP Computer Science Principles curriculum is your Create tasks. As the name suggests, it's meant to be creative. You should take the programming skills you have learned and use them to build something which is your own idea.

Collaboration: You are encouraged to collaborate with a friend in brainstorming and planning your project. It is possible to work on the code together, but if you do, you need to divide the work so that each of you writes a different part of it - read the official guidelines closely before you start. It is probably easier if you just write your own, and ask your friends for feedback or debugging help.

How Complex Should it Be?: These projects can be surprisingly simple. Part of the grade is determined by how well you explain your project, using a video and some short answer questions. The most important thing is that your code must include an algorithm and an abstraction.

What Language?: Please use p5js! You can use something else only if you have another favorite language that you have more experience in, and you think your project will be more interesting if written in that language.

Happy Creating!

Chapter II

Study the Task Requirements

Check out the handouts and preparation resources available in [this folder](#).

II.1 Form a pair

Choose a partner to work with. They can help you with brainstorming & planning your project code, and you should work together on grading the example projects.

II.2 Understand what you will turn in

The Create Task includes a few parts beyond just code. By the end of the year you'll turn in following:

- A PDF copy of the program code. You make do this by copying your code into Pages or Google Docs and saving to PDF from there.
- A screencast video demonstrating how your program runs.
- Written responses.

It is important to remember that although you can work with a partner, you must have specific parts of the code that you can claim as your own.

II.3 Read the Manual

Reference the [Official College Board guide](#), page 9-13, and also the Code.org "[Create Part - Survival Guide](#)". This has great information about how to think about the algorithms and abstractions in your code.

II.4 Grade practice examples

Choose three or more of the examples from [this folder](#).

Using the [2019 Scoring Rubric](#), give the example project a score out of 8 points.

II.5 Check your grading against the key

Use the answer keys in [this folder](#) to check your understanding of the scoring guidelines against the answer key. Talk about the answers until you are sure that you understand why each example received the grade that it did.

Chapter III

Brainstorm your own project

Do you have an idea for a project you would like to create?

Here are some types of projects that you could create:

- An interactive game, even if it's a simplified version of a more complicated game.
- A tutorial that demonstrates how to do something, or teaches the best strategies for some game.
- An animation that tells a story or visualizes something beautiful or mathematical.
- A widget that helps you perform a calculation or look up information.

Chapter IV

Coding

This project is meant to take about **12 hours** of class time to complete the whole project, including the video and short answer questions. You may use more time at home if you want to.

It's normal if your attention span for *one* topic is only about 2 hours. You could work on the project for 6 weekends, using 2 hours of the class time each time - and use another 1 or 2 hours that you are here to research APCSP vocabulary words and work through the other class assignments.

1. Week 1: Grade the example projects and learn what you need to do to get a 5. Brainstorm ideas and talk to a buddy about which of your ideas will work best.
2. Week 2: Use a pencil and paper, or a whiteboard, to plan out the different parts of your code. Then write as much of the code as you can.
3. Week 3: Open your first draft and check if it has an algorithm and abstraction. Keep working on it until you have a simple program that runs without errors.
4. Week 4: Keep working on your code to make the program a little better, and to make sure that the code is well organized and readable. Explain what your algorithm and abstraction are to a mentor, as practice before explaining it to the College Board.
5. Week 5: Put your code into a PDF format and highlight the algorithm and abstraction parts. Start answering the short answer questions.
6. Week 6: Record the video of your program running, and finish your short answer questions or edit them.

Chapter V

Resources for Finishing the Project

[Written Response Questions Template](#)

[How to record your screen on Mac](#) - our computer lab is on High Sierra, not Mojave, so follow the Quicktime Player instructions.

Easiest way to save your code as a PDF:

1. In the p5js editor, use shift+tab to auto-format your code. Proper indentation and spacing will make it look better to the graders.
2. Use command+A to highlight all your code, and then command+C to copy it.
3. Open a new document in Google Docs, and paste your code there.
4. Use the menu File -> Download As -> PDF Document. Voila!
5. Open the PDF in Preview. Use the menu option Tools -> Annotate -> Rectangle, or Oval, to draw shapes around your algorithm and abstraction sections.