AP Create PT Prep

This short unit prepares students to complete the AP® Create Performance Task (PT). Students will have learned the skills and concepts necessary to complete the task in previous units and will even have seen components of the task itself. This unit fully explains all components of the task and walks students through completing and submitting it.

Chapter 1: Create PT Prep

Week 1

Lesson 1: Create PT - Review the Task

This lesson contains a series of activities you can use to help students familiarize themselves with Create Performance Task, how it is scored, and some example tasks provided by the College Board.

Lesson 2: Create PT - Make a Plan

This lesson uses the Create PT Survival Guide as the backbone for a series of activities to ramp up to doing the actual Create PT. It contains activities to help students understand the algorithm and abstraction requirements of the task, as well as activities to help them narrow down and brainstorm ideas for their actual project.

Week 2

Lesson 3: Create PT - Complete the Task (12 hrs)

Project

The lesson includes some final reminders and guidelines for completing the Create PT before officially starting. For a total of 12 class hours, you will work on your project with only types of teacher support allowed (essentially: Advise on process, not ideas). You may also work with a collaborative partner in *in development of you program* - written responses must be done on your own.

Chapter Commentary

Key Concepts and Pedagogy

Create PT After Unit 5 Chapter 1: Students are prepared to complete the Create PT after Unit 5 Chapter 1. If time allows, you may opt to wait until after Unit 5 Chapter 2.

Highlight Ambiguities with Activities: It can be challenging to understand the expectations of the Explore PT and the way it is graded. This unit solves this problem through hands-on activities designed to bring these ambiguities to the surface. By grading sample projects or assessing potential project choices in a group setting, students are able to understand what is expected of them and ensure their projects exceed this bar.

Survival Guides: The most significant resources in this unit are the Survival Guides created for each task. In addition to including activity guides for preparation activities, they include instructions for how students can identify good project choices and plan their allotted class time. Checklists for each task help students track their progress as they complete them.

Practice Earlier in the Curriculum: Students learn the skills and concepts necessary to complete the performance tasks throughout the course. Each unit includes at least one Practice PT or Rapid Research project specifically designed to prepare students for different elements of the tasks.


This curriculum is available under a
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes,**contact us.**

Lesson 1: Create PT - Review the Task

Overview

This lesson contains a series of activities you can use to help students familiarize themselves with Create Performance Task, how it is scored, and some example tasks provided by the College Board.

Students review the Submission Requirements and Scoring Guidelines for the Create PT. Subsequently they review three example scored Create PT submissions with commentary to better understand how the Submission Requirements and Scoring Guidelines are used together. In a wrap-up conversation they identify a piece of advice, a "gotcha", and a remaining question they have about the Create PT.

Note: Much the sample tasks, scores, and commentary on scoring shared in this lesson come directly from the College Board. Code.org's commentary is noted where applicable.

Purpose

The Create PT is in many ways straightforward: you complete a self-directed programming project and respond to prompts about your program and process. As you dig into the details of the task, however, you quickly come across some of the nuances of individual components of the task and how they're scored. This lesson is designed to introduce what these nuances are, and begin to provide some answers to the questions that will inevitably arise. Keep in mind that the next lesson provides a more structured set of responses to those questions, and so today students are just diving in to what the task looks like.

Agenda

Review the Task

- **Introduce the Create PT**
- **Review Create PT Submission Requirements and Scoring Guidelines**

Review Scoring Guidelines and Sample Tasks

- **Create PT Sample Response C**
- **Create PT Annotated Sample C (score: 7/8)**
- **Create PT Annotated Samples F (5/8), I (3/8), and J (1/8)**
- **Optional - Review Grumpy Cat Exemplar Create PT**

Wrap Up

- **Create PT: Advice, Gotchas, Questions**

(Optional) AP Digital Portfolio Setup

- **Tech Setup - AP Digital Portfolio, Making PDFs, and**

View on Code Studio

Objectives

Students will be able to:

- Describe the major components of the Create PT
- Describe how the Create PT Scoring Guidelines will be used to assess the task
- Evaluate sample Create PT submissions by applying the scoring guidelines
- Identify remaining questions about the Create PT

Preparation

- [ ] Print or prepare to distribute digital copies of **AP CSP Performance Task Directions for Students - Resource**
- [ ] Briefly review all of the graded sample Explore PTs included in the lesson plan
- [ ] **Optional:** While not provided to students, the documents below let you see how each written response was graded in all 10 sample submissions provided by the College Board. These may help you better understand expectations for each question and answer student questions.
  - [ ] **Create PT - Response 2b - All Samples**
  - [ ] **Create PT - Response 2c - All Samples**
  - [ ] **Create PT - Response 2d - All Samples**

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **Create PT - Response 2b - All Samples** Make a Copy
- **Create PT - Response 2c - All Samples**

Videos

Make a Copy

- **Create PT - Response 2d - All Samples** **Make a Copy**

For the Students

- **AP CSP Performance Task Directions for Students** - Resource
- **Create PT - Scoring Guidelines 2019** - College Board Doc
- **Create - Video: Sample C**
- **Create - Written Responses: Sample C**
- **Create PT Annotated Sample C 2017 (7/8)** **Make a Copy**
- **Create PT Annotated Sample F (5/8)** **Make a Copy**
- **Create PT Annotated Sample I (3/8)** **Make a Copy**
- **Create PT Annotated Sample J (1/8)** **Make a Copy**
- **Grumpy Cat Exemplar Create PT Written Responses**
- **Grumpy Cat Exemplar Create PT Code**
- **AP Digital Portfolio Student Guide** - Resource
- **AP Digital Portfolio Access** - College Board Site

Teaching Guide

Review the Task

Introduce the Create PT

Remarks

> Today we're going to start looking more deeply at the Create PT, focusing specifically on understanding:
>
> - The different components of the Create PT
> - How the task will be scored
>
> Don't worry, you already have much of the knowledge and skills you need to do well on this task. The hardest part might be just understanding what is required of you.
>
> First, we'll quickly read the task description and look a some examples and how they were scored.

Review Create PT Submission Requirements and Scoring Guidelines

Students can find links for all these activities in Code Studio. Here is what they can see:

Code Studio levels

- Levels
- **1**

Student Instructions

View on Code Studio

Unit 6: Lesson 1 - Create PT Prep - Reviewing the Task

Overview

This lesson contains a series of activities you can use to help students familiarize themselves with Create Performance Task, how it is scored, and some example tasks provided by the College Board.

Resources

College Board Documents

- **AP CSP Performance Task Directions for Students** - College Board Student Handout
- **Create Performance Task - Scoring Guidelines 2018**

College Board Create PT Samples

- **Create - Video: Sample C**
- **Create - Written Responses: Sample C**

Annotated Create PT Samples

- **CB Create PT - Sample C (7/8)** (copy as **MS Word**, **Google Doc**)
- **CB Create PT - Sample D (7/8)** (copy as **MS Word**, **Google Doc**)
- **CB Create PT - Sample I (2/8)** (copy as **MS Word**, **Google Doc**)
- **CB Create PT - Sample J (4/8)** (copy as **MS Word**, **Google Doc**)

Code.org Create PT Exemplar

- **Grumpy Cat Code**
- **Grumpy Cat Written Responses**

(optional) Digital Portfolio Setup AP Digital Portfolio Student Guide - College Board Handout AP Digital Portfolio Access - College Board Site

Continue

Distribute: Students should each get printed or digital copies of **AP CSP Performance Task Directions for Students - Resource** and the **Create PT - Scoring Guidelines 2019 - College Board Doc**

Prompt: Read and then discuss with a partner (1) the "Submission Requirements" section on pages 10-11, and (2) the Scoring Guidelines. For the Scoring Guidelines you can focus only on the first 3 columns for now: "Reporting Category", "Task", "Scoring Criteria". We'll dive into the decision rules later. Just get familiar with these documents.

After reading discuss with a partner:

- What will you **actually** be turning in to the College Board?
- What are you hoping will become more clear after looking at example projects?

Discuss: Give students time to read the pages, in pairs or individually, and then discuss both questions with one another. The first one is more important for now.

Remarks

> Hopefully in your reading you concluded that for the Create PT you'll need to submit:
>
> - Video showing your program running
> - A PDF of your program code
> - Written responses
>
> You should also have noticed:
>
> - The Scoring Guidelines provide specific guidance on how each part of the task will be graded
>
> I'm sure that right now you have a lot of questions about what this task will look like and how it will be scored. Before we answer them, let's look at some examples first.

Review Scoring Guidelines and Sample Tasks

Create PT Sample Response C

Distribute: Provide pairs of students copies of Create Sample C from the 2018 sample set (links available on student page

Teaching Ti

Students can find all the links they need on this page. (If you are viewing this from within the lesson plan click **Student Overview** to see the student view of the page).

Warning: We **have included links** on this student page to the the scoring commentary for the sample tasks. If you you want students to try applying the scoring guidelines to tasks **without knowing ahead of time** you'll have to invoke some kind of honor system or strategy to have them hold off looking.

Deprecated - old content from this page

On Dec. 13, 2017 we updated this page with new resources and links. If you're looking for the old stuff, it is copied below, but should be considered out of date.

Background

Review the Create performance task, scoring guidelines, and sample tasks.

Lesson

- Review the Create Performance Task.
- Review Create PT Samples Tasks and Scoring Guidelines

Resources

College Board Documents

- Create PT - **Guidelines before starting**
- **Create Performance Task** - Task Description
- **Scoring Guidelines** - Create Performance Task
- **Create PT Samples:**
- Sample C (**video** | **written responses**)
- Sample J (**video** | **written responses**)
- Sample D (**video** | **written responses**)
- **Scoring Commentary**
- For all samples - **Sample Create PTs Scoring Commentary**

Other Resources

- Create Performance Task - Guidelines Before Starting (**PDF** | **DOCX**)

for this lesson).

- **Create - Video: Sample C**
- **Create - Written Responses: Sample C**

Prompt: This is a raw student submission - exactly what the student uploaded for their computational artifact and written responses. Read it to yourself first. Then with your partner spend a few minutes reviewing it. Be ready to share out the following answers.

- Did anything surprise you in looking at this sample?
- Do you think this scored well based on what you know about the scoring guidelines?

Discuss: Ask partners to spend a couple of minutes specifically discussing the prompts above. Then have the whole class quickly share the results of their discussion.

Teaching Tip

Understanding the Prompts: While not provided to students, the documents below let you see how each written response was graded in all 10 sample submissions provided by the College Board. These may help you better understand expectations for each question and answer student questions.

- **Create PT - Response 2b - All Samples**
- **Create PT - Response 2c - All Samples**
- **Create PT - Response 2d - All Samples**

Write Questions on the Document: If students have printed copies of the **AP CSP Performance Task Directions for Students - Resource** you may wish to encourage them to underline, circle, or otherwise mark the questions they have in the following prompts.

Create PT Annotated Sample C (score: 7/8)

Remarks

> Sample C actually received a 7/8 score. Let's look at the student response side-by-side with the scoring guidelines **and** the actual AP scorer's notes to see why.

Distribute: The **CB Create PT - Sample C (7/8)** (links available on student page).

Prompt: With your partner look over this annotated version of the sample to see how each row of the scoring guidelines was applied. **You should be reading specifically to answer any of the questions you had about the task earlier.** After looking it over we will discuss:

- What characteristics of this response made it score well?
- Why specifically did this submission not earn Row 6?
- What questions do you still have about the Scoring Guidelines or Task description?

Discuss: Ask partners to spend a couple of minutes specifically discussing the prompts above. Then have the whole class share the results of their discussion.

- Where possible call out ways that the discussion is answering questions raised earlier in the class about the Submission Requirements or Scoring Guidelines.
  - The specific algorithm requirements are fairly nuanced. It's very possible to write a good complex algorithm that doesn't actually feature the **selected** algorithm with two **included** algorithms.
  - The rest of the responses are very precise in their language. They use the language of the prompts and make it very easy to find the information the prompts are asking for.

Discussion

Goal: Aim to keep this discussion relatively short. Assure students you're intending to log their questions and they will be addressed through the lesson.

Students should leave this discussion knowing they will submit:

- video of their code running
- written responses
- PDF of program code

They should also know the Scoring Guidelines:

- contains 8 rows, each worth 1 point
- sometimes several rows apply to one written response to pick out specific aspects

Students are not, however, expected to fully understand the nuances of the task or scoring.

Student Samples

The student samples used in this activity come directly from the **AP Central website** which shows in separate documents: student samples, scoring guidelines, and scoring notes. You can direct students there to find the samples, or look at others if you like.

Later in the activity we provide "annotated" versions that merge all three of these things together into one side-by-side view.

Create PT Annotated Samples F (5/8), I (3/8), and J (1/8)

Remarks

Let's now take a look at some other samples.

Distribute: Provide pairs of students copies of the Annotated Create PT Samples F, I, and J (student links on code studio)

- **Create PT Annotated Sample F (5/8)**
- **Create PT Annotated Sample I (3/8)**
- **Create PT Annotated Sample J (1/8)**

Prompt: With your partner look at these samples - you can pick which to look at first. As you review this task with a partner ask yourself:

- Where and how specifically did this fall short?
- Was there one major problem that caused ripple effects through the scoring?
- Or were there several smaller issues?
- Try to point out specific aspects of the Scoring Guidelines or Submission Requirements.

Discuss: Ask partners to spend a couple minutes specifically discussing the prompt above. Then have the whole class share the results of their discussion. Where possible call out ways that the discussion is answering questions raised earlier in the class about the Submission Requirements or Scoring Guidelines.

Optional - Review Grumpy Cat Exemplar Create PT

Review: The Code.org curriculum team felt students could benefit from seeing an exemplar Create PT project in which they could see and even edit the entirety of the program code. The links below are to an exemplar Create PT submission and program code we believe would earn full credit on the 2018 Create Performance Task.

- **Grumpy Cat Exemplar Create PT Code**
- **Grumpy Cat Exemplar Create PT Written Responses**

Wrap Up

Create PT: Advice, Gotchas, Questions

Prompt: Based on the examples that you saw today write down on separate post-its / scratch piece of paper

- The number one piece of advice you have for the Create PT
- One "gotcha" to look out for
- One question you'd still like answered about the Create PT

Discussion

Goal: Students should understand from this example that the Scoring Guidelines are in many ways as important as the task description. The responses in this sample not only match the task description but address the particular "gotchas" of the scoring guidelines.

This response didn't earn Row 6 even though they wrote a fairly complex set of algorithms. Call out that the very specific algorithm requirements are something that will be covered in detail in the next lesson.

Students may still have questions about the individual prompts or scoring guidelines. Encourage them that you'll look at more examples which may help clarify.

Discussion

Goal: Students should be gaining comfort with the structure of the task and scoring guidelines at this point. Since each of the sample tasks provided missed some points it provides a good opportunity to dive into those components of the scoring guidelines.

Here are some key ideas to take from each of the samples.

Sample F:

- For Row 2 the response only focuses on two problems and their solutions. This does not meet the incremental (step by step) development process requirement. The response also does not explain if the problems were solved using an iterative development process related to feedback, testing, or reflection.
- For Rows 4 and 5 the response does get the point for the **selected** algorithm (Row 4) but neglects to get the point for Row 5 because the response does not define how the algorithm relates to the program as a whole.
- For Row 6 the response does not define the **included** algorithms. As a reminder, there should be three total algorithms explained (**selected** algorithm and two **included** algorithms).

Sample I:

- For Row 2 the response does not receive credit for similar reasons to Sample F.
- Row 3 is a little tricky. At first glance it appears that two problems are defined and solutions given. However, one problem is not from the program development process, but instead a problem in the design process. For example, discussing how difficult it was to set up the layout of an app is not a good choice for this prompt because that's a design challenge. Discussing the difficulties in debugging a score-setting function would be an appropriate choice.
- For Rows 4 and 5 the response earns the points because the **selected** algorithm is an algorithm that

Discuss: Have students share their answers with a partner. Then have them place their responses on the board somewhere where they can be seen.

Once answers are on the board quickly report back to the group the patterns or trends that you're seeing in their responses.

Remarks

> Next time we meet we're going to look more deeply into the Create PT, using the three questions you just answered. We'll look closely at the algorithm and abstraction requirements and help you decide what kind of project to make. We'll also talk about strategies for avoiding many of the "gotchas" you identified in this lesson. Finally, we'll take time to address any remaining questions you have about the task.

(Optional) AP Digital Portfolio Setup

Tech Setup - AP Digital Portfolio, Making PDFs, and Videos

At some point students need to setup their AP Digital Portfolio to officially submit your performance tasks and to sign up for the exam.

Doing that setup and navigating around the digital portfolio will take a little bit of time.

The resource you need for that is primarily the: **AP Digital Portfolio Student Guide - Resource** There are also several tools you should be familiar with in order to create the necessary PDF documents and Video screen captures that you need to submit. We provide links and some other instructions around tech-related things in the first level for this lesson.

uses math and logic and the response explains the purpose of the algorithm in the context of the program.
- However for Row 6 the response failed because to receive this point, the **selected** algorithm must be clearly defined along with two **included algorithms** that can function independently. There must be at least three distinct algorithms defined and explained within the response.
- For Rows 7 and 8 the code segment is not an example of an abstraction such as a function or a list. The student selected two lines of code and explained how they worked. The student did not explain how the abstraction managed complexity.

Sample J:

- Building a project using mostly event handlers (onEvents) can be a fun way to construct an interactive app. However, these types of apps will not satisfy the Create Performance Task requirements unless there are complex algorithms in addition to the onEvents.
- For Rows 2 and 3 the response details how the game works and design challenges faced instead of explaining the process of creating the program for the app step by step (incremental development) and how parts were improved through testing, reflection, or feedback (iterative development).
- For Row 4 in this case the AP Scorer is looking to see if an onEvent can count as an algorithm. A collection of onEvents does not count as a single algorithm. Each individual onEvent only contains a single instruction which disqualifies it as an algorithm. To receive the point for Row 4, the algorithm must contain sequencing (more than one step), selection (if-statement), or iteration (for loop).
- For Row 5 the code segment is not an algorithm and does not use math or logic, so no point is awarded.
- It is possible to receive a point for Row 6 even if Row 5 is not awarded a point. However in this case no point is awarded because the two **included** algorithms are not defined or explained.

Teaching Tip

Do Not Remix This Program: This exemplar should be useful for students to better understand what a Create PT submission might look like. That said, copying or remixing this project is likely to end up getting students flagged for plagiarism. Students should use this project for inspiration but should be reminded not to use or remix the code for use in their actual Create PT. If you need, remind students that both tens of thousands of students and the College Board have access to this program.

Discussion

Goal: The next lesson is designed to address these three

Teaching Tip

Pick the right time to do this tech setup. We've included the resources you need both in the AP Create prep lessons and the AP Explore prep. The purpose is to have a place to go for quick links to things like setup guides and other tools.

Code Studio levels

- Levels
- 2

Student Instructions

View on Code Studio

Tech Setup and Tools for the AP Performance Tasks

Teaching Tip

For the Teacher

- Here is the **AP Digital Portfolio: Teacher User Guide for AP Computer Science Principles**
- You will need to follow these instructions to setup your class and to approve students

Background

You need to setup your AP Digital Portfolio to officially submit your performance tasks and to sign up for the exam. There are also several tools you should be familiar with in order to create the necessary PDF documents and Video screen captures that you need to submit.

Resources and Quick Links

- **AP Student Guide for setting up digital portfolio** (PDF)
- **AP Digital Portfolio**
  - **AP Central web page about the digital portfolio**
- **Code Print** - useful tool for preparing PDF of code for Create PT

More Details -- Table of contents

- **AP Digital Portfolio Setup** (Create & Explore PT)
- **Making PDFs for Written Responses** (Create & Explore PT)
- **Making PDF of Program Code** (Create PT only)
- **Making a Video Screen Capture** (Create PT required, could also use for Explore PT computational artifact)

AP Digital Portfolio Setup

Goal: Students should be aware of the Digital Portfolio and how to access it. They should know what's there and be familiar with the basic mechanics of uploading and submitting their projects.

If your students have not done this yet, they will need to register themselves with AP digital portfolio in order to upload their projects.

Follow College Board Instructions to Setup Portfolio

- Here is the **Student Guide for the digital portfolio** . Follow these instructions to get setup.

The digital portfolio and guide contains a few helpful other things students should know about such as:

- Guidance about how to create a PDF
- Templates for the written prompts
- Ways to save drafts of written responses on the site and come back to it

---

Making PDFs for Written Responses

You are required to make a PDF of your written responses to prompts. It's recommended that you use the College Board templates for filling out your responses. At some point you will have your written responses in a word processing document such as Microsoft Word, Google Docs, or Pages.

What follows is copied from the AP Student Guide for the digital portfolio

How to make a PDF

- Recent versions of applications like: Word, PowerPoint, Pages, and Google Docs, have built-in features that allow you to save or export your file as a PDF. Instructions are provided below.
- If your software does not have a PDF option, visit the Adobe site and learn more about whether Acrobat from Adobe Systems can convert your document to PDF.
- You are responsible for ensuring that your file is properly formatted and readable. After you have created your PDF, be sure to check it by opening and reviewing your PDF in Adobe Reader, a free application that can be downloaded from the Adobe site.

Microsoft Office (Word, PowerPoint)

- In Word, Powerpoint, and other Microsoft Office programs you will "Save as PDF." Visit the Microsoft Office support page for more information about "Save as PDF." To save a Word or Powerpoint document as PDF:
  - Open your Word or Powerpoint document.
  - From the top menu select "File," and select "Save As."
  - In the dialog window, go to the drop-down menu for "Save as type," and select "PDF."
  - Click "Save."

Google Docs

- In Google Docs, you will "Download as" PDF. Visit the Google support page for more information on "Download a file." To download a Google Doc as a PDF:
- Open your Google doc
  - From the top menu select "File," and select "Download as," and select "PDF Document (.pdf)"

Pages

- In Pages, you will "Export to" PDF. Visit the Apple support page for more information or follow the steps below:
- Open your Pages document.
- From the top menu select "File," and select "Export to," and select "PDF."
- In the dialogue window select "Best," image quality.
- Choose a destination for the export and click "Export."

Making PDF of Program Code (for the Create PT)

You need to make a PDF of your code and you**also** draw an oval and rectangle onto the PDF to highlight certain parts. There are a few options for this.

- Our recommendation: Use **CodePrint** - a tool for doing everything from the browser.
- Option 2: Make a PDF of the Code, then Edit the PDF using a PDF editor to draw shapes
- Option 3: Copy/Paste Code into a Word (or Google) document and add shapes there to produce PDF.

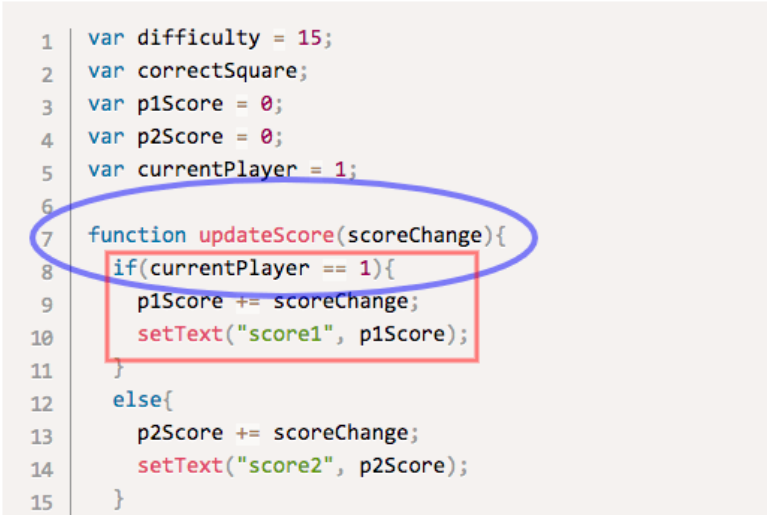
Details: how to make a pdf of your code

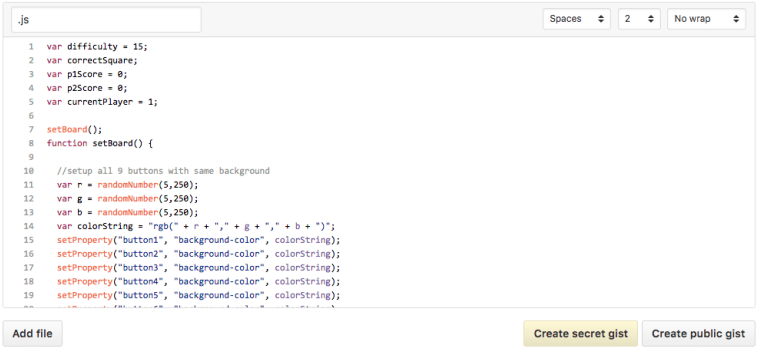
Step 1 - copy your code in App Lab

- Switch App Lab into text mode
- Select all the code(highlight all with your mouse or Ctrl+A)
- Copy it (Edit -> Copy, or Ctrl+C)

Step 2 - paste the code into a page or doc for printing

- **If using CodePrint**
  - This tool lets you draw rectangles and ovals over a pretty-ified version of the code (diagram at right)
  - If you can print a PDF from the browser, this should be all you need.
- **Other options**
- **Option: Github Gist** -- **GitHub Gist** is a tool designed to let you quickly share code. We can use it to quickly print as well.
  - Go to **GitHub Gist**
  - Paste your code into the code area (the large open area with line numbers)
  - Optional: In the filename box type .js -- this forces the box to recognize the code as javascript
  - Click "Create Secret Gist" - this will save the code to a new page anonymously
  - From your Browser choose "File -> Print" and use your computer's option to print to PDF.
- **Option: use a word processor Google docs or MS Word**
  - This option is fine but you won't get line numbers next to your code which can be convenient.
  - If you choose this option you should add your annotations (rectangle and circle) here in the word processor.





How to add Ovals and Rectangles to a PDF

If not using **CodePrint** you'll need to add ovals and rectangles to the PDF of your code.

Windows

- You need to install Adobe Acrobat (see the AP guide for students)
- Open the PDF in Acrobat and add annotations

Mac

- The built in Preview App allows you to add rectangles and ovals directly
- With PDF open in Preview go to Tools -> Annotate -> Rectangle for example.

Making a Video Screen Capture

Students are required to make at least one video that is a "Screen capture" of themselves using the program they wrote for the Create PT.

How To Make a Screencast

If you have not made any screencapture videos in class to this point students may ask how to do it. You will need to use screen capture software. Here are two good options.

- Online/Web: **Screencast-o-matic** (may require download depending on browser/plugins)
- Windows / Mac: **Jing from Techsmith** (requires download)

Standards Alignment

Computer Science Principles

▸ **1.1** - Creative development can be an essential process for creating computational artifacts.

▸ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.

▸ **5.1** - Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).

This curriculum is available under a
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes,**contact us.**

Lesson 2: Create PT - Make a Plan

Overview

This lesson uses the **Create PT Survival Guide** as the backbone for a series of activities to ramp up to doing the actual Create PT. It contains activities to help students understand the algorithm and abstraction requirements of the task, as well as activities to help them narrow down and brainstorm ideas for their actual project.

The lesson concludes by providing students with resources to make a plan to complete the task staring in the next lesson.

Purpose

There are no new CS concepts covered in this lesson. It is a review of the processes and requirements of the Create Performance Task before students begin working on it individually.

Agenda

Getting Started (5 mins)

Activity (70 mins)

- **Introduce The Create PT Survival Guide**
- **Is It a Good Algorithm? (page 2)**
- **Activity: Does It Count? - Algorithm Edition (pages 2-4)**
- **Is It a Good Abstraction?**
- **Activity: Does It Count? - Abstraction Edition**
- **Brainstorm Project Ideas**

Wrap-up (10 Minutes)

- **Making a plan**

View on Code Studio

Objectives

Students will be able to:

- Describe the elements and purpose of the Create PT
- Describe the scoring guidelines for the Create PT
- Evaluate sample Create PT components by applying the scoring guidelines

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **[KEY] Create PT - Survival Guide 2018/2019** **Make a Copy**

For the Students

- **Create PT - Survival Guide 2018/2019** **Make a Copy**
- **Create PT Check-In Organizer** - High Resolution PDF (**download**)
- **AP CSP Performance Task Directions for Students** - Resource
- **Create PT - Scoring Guidelines 2019** - College Board Doc

Teaching Guide

Getting Started (5 mins)

Prompt: Based on our review of the Create PT yesterday...

- What are the main things you have to do for the Create PT?
- What should you do first?

Discuss: Let students call out the things they remember. Make this a quick review just to remind them of what was covered in the previous lesson.

A few of the things you need to do:

- Write a program
- Individually write code for an algorithm and abstraction
- Make a video of your code running
- Answer written responses
- Make a PDF of your code

What should you do first?

- Pick and scope down your project

Discussion

Goal: The purpose of this discussion to warm up students' brains and recall the elements of the Create PT. Students should start thinking about choices in light of what they have to do for the Create PT, rather than simply out of interest or "coolness." Hopefully, the two go hand in hand.

In particular we'll looking to understand the algorithm and abstraction requirements so that students select appropriately scoped projects.

Teaching Tip

Use the Grumpy Cat Exemplar: Consider referring back to the "Grumpy Cat" exemplar Create PT included in the previous lesson either as part of this review or as a reference during the remainder of the lesson.

- **Grumpy Cat Exemplar Create PT Code**
- **Grumpy Cat Exemplar Create PT Written Responses**

Activity (70 mins)

Introduce The Create PT Survival Guide

Remarks

> Today we will use the Create PT Survival Guide to dig in a little deeper with the Create PT. The beginning of the packet has a number of quick activities that help get us in the right mindset for thinking about and doing the task so you are ready to hit the ground running.
>
> The guide is intended to be useful **throughout** the entire process of completing the actual Create Task as well.

Distribute: **Create PT - Survival Guide 2018/2019**

Read Overview (page 1): Have students read the **Create PT Overview** section of page 1.

- The information about the task itself should not be news if you've already reviewed the Create PT
- Focus on the suggested process and timeline on page 1

Algorithms on the Create PT (20 mins)

Is It a Good Algorithm? (page 2)

Prompt: The Create PT requires you to write and describe an algorithm in your program. Based on what we learned yesterday looking at scored samples, and the criteria listed here, what are the characteristics of that algorithm?

- Give students time to read
- Allow students to chat with a neighbor
- **Highlight the evaluation criteria** specifically listed

Teaching Tip

The **Is It a Good Algorithm** page contains information useful for the **Does it count?** activity. Some of the advice like the "parent and two children" (**selected** and **included**) portion of algorithms might be hard to understand in the abstract. You could still review it first and then revisit after going through the examples.

Alternatively, you could just dive into the **Does it count?** activity and tell students that all the information they need to make judgements about the algorithm code segments is contained page 2.

in the Survival Guide
- It is not necessary to do a whole group discussion here - this is just a kick off.

Transition: To get our brains moving we're going to look at some submissions for the algorithms and discuss whether they're good choices.

Activity: Does It Count? - Algorithm Edition (pages 2-4)

Prompt: Using what we just discussed about your algorithm choices and the scoring guideline provided, determine for each algorithm whether it would get the point and why.

Have students individually take 5-10 minutes to look at the set of selected algorithms and respond about whether those algorithms would earn a point, and why.

Discuss and Review Have students share and compare their responses with a classmate. Afterwards lead a discussion based on the patterns you see. You'll likely want to review the following points from the Survival Guide

Discussion

Goal: We want students to understand that in looking at the code an AP reader is looking for very specific things - it's a checklist. Familiarity with that checklist helps when planning your own project.

Don't worry or wallow too much in the gray areas of some of the examples - they are intentionally borderline to show where the border is. Students should try to be safely over that border.

Use the **Create PT Survival Guide - KEY** for commentary on individual algorithms.

- The "parent and two children" model for a complex algorithm - (main + 2 sub-algorithms)
- What counts as mathematical and logical concepts

Abstraction on the Create PT (20 mins)

Is It a Good Abstraction?

Prompt: The Create PT requires you to write and describe an abstraction in your program. Based on what we learned yesterday looking at scored samples, and the criteria listed here, what are the characteristics of that algorithm?

- Give students time to read
- Allow students to chat with a neighbor
- **Highlight the evaluation criteria** specifically listed in the Survival Guide
- It is not necessary to do a whole group discussion here - this is just a kick off.

Transition: To get our brains moving we're going to look at some submissions for the abstraction and discuss whether they're good choices.

Teaching Tip

We make a big deal about not selecting `onEvent` as an abstraction because it is the major pitfall of students using event-driven programming like we use in App Lab. While there is technically a function involved, it is not a named function and not one the student decided they needed to manage complexity. Using onEvent, some like to say, is not a "deliberate act of abstracting" it's just how the language deals with event-handling.

There are other abstractions allowed - like data abstractions, using lists in particular - but it's so easy and common practice to make functions in app lab to handle certain problems it's low hanging fruit for the Create PT.

Activity: Does It Count? - Abstraction Edition

Prompt: Using what we just discussed about your abstraction choices and the scoring guidelines provided, determine for each abstraction whether it would get the point, why, and whether you can determine whether it manages complexity.

Give students individually 5-10 minutes to look at the list of abstraction selections and write down their judgements about whether it should earn the point, why and whether it manages complexity.

Discuss: Have students share and compare their responses. Afterwards lead a discussion based on the patterns you see. You'll likely want to review the following points from the Survival Guide

- `onEvent` does not count as a student-developed abstraction
- Any function you defined and wrote yourself can earn the point for row 7

- To justify that a function helps manage complexity you must be able to demonstrate how/where you call that function from two different places in your code or, if it has a parameter, call it with different values.

"Narrow it Down" (15 mins)

Prompt: Read the "Narrow It Down" section of your survival guide. Be ready to discuss:

- What are the key takeaways from this section?
- What questions do you have?

Discuss: Have students share their thoughts. The most important points to note:

- The written responses are the most important part of the Create PT.
- It's OK to submit an incomplete project so long as it has a working feature you can show in your video and contains an algorithm and an abstraction
- Most ideas can and should be narrowed down before you start
- You shouldn't be doing a lot of work in Design Mode.
- You need to worry about the code that makes your program work, not making your initial screens perfectly lined up or attractive.

Practice Narrowing It Down

Prompt: With a partner go through these three example project proposals. For each one practice narrowing down the features of the project and identifying the core algorithm.

Give students 5-10 minutes to look at the three projects and start to jot down their own ideas.

Discuss: Have students share and compare their responses. Afterwards lead a discussion based on the patterns you see. You'll likely want to review the following points from the Survival Guide:

- Many projects have sub parts, each of which could stand on its own as a PT
- You should be able to easily see an algorithm opportunity within at least one of the sub parts -- if you can't, not a good choice.
- For any project idea it **should** be relatively easy to scope it down to one or two things that will be totally acceptable for the Create PT

Discussion

Goal: Similar to the algorithm section, we want students to understand that in looking at the code an AP reader is looking for very specific things - it's a checklist. Familiarity with that checklist helps when planning your own project.

Again, don't worry or wallow too much in the gray areas of some of the examples - they are intentionally borderline to show where the border is. Students should try to be safely over that border.

Use the **[KEY] Create PT - Survival Guide 2018/2019** for commentary on individual abstractions.

Discussion

Goal: Understand it doesn't have to be a big project; The create PT is about demonstrating something you already know how to do.

The biggest thing we're trying to guard against is students' eyes being bigger than their stomachs. We want to encourage students to be creative and start build whatever they want, but temper that with the realities of the Create PT...

- It doesn't need to be a big project
- Your job is to demonstrate that you know how to program something and identify certain aspects of it.
- There are no points for coolness or prettiness
- If you want to do something big, just get it started for the Create PT and come back to it afterward.

Teaching Tip

You might consider doing a **jigsaw** here. Have students pick **one** of the scenarios to do and come together to discuss afterward.

Bring It All Together (15 mins)

Prompt: Read the "Bring It All Together" section of your Survival Guide. Be ready to discuss:

- How should I go about selecting my project?

Discuss: Have students share their thoughts. The most important points to note:

- You don't actually have that much time to work!
- When you start, you should have an idea about what the algorithm will be.
- Start with a narrowly scoped project, start working right away on the core parts of it.
- Don't try to learn new programming skills during the PT - do something you know how to do now.
- Get to the written responses as quickly as you can.

Brainstorm Project Ideas

Prompt: Come up with two example project ideas. List some simple information about each project so that a partner can give you some feedback on your idea.

Give students 5-10 minutes to look at the three projects and start to jot down their own ideas.

Share: Have students share their first-draft ideas with a partner and list feedback on the right-most column of the activity. Afterwards they can confer with a partner about their ideas.

Discuss: Have students share and compare their responses. Have student think about starting to pick their own projects. When deciding on a project the answer to these questions should all be "yes".

- You know what your algorithm is probably going be?
- Can you do all the programming for this in about 6 hours?
- Can you get to the algorithm within first 2 hours?
- You feel like you know how to do most of the programming for this **right now**?

Teaching Tip

Do you need to plan for abstraction? . You probably don't need to plan for abstraction explicitly - if following the practices of this course, something will emerge that you can use.

Discussion

Goal: Students should exit this brief activity with (1) a basic idea of what they're going to do for their project and (2) confidence that they can do it.

Wrap-up (10 Minutes)

Making a plan

- Turn to **pages 11 and 12** of the survival guide
- The pages are there to help you plan for doing your own Create PT.
- The **Create PT Process/Check-in Organizer** is a place to take and categorize your notes and thoughts as you make your project.
  - This **Create PT Check-In Organizer - High Resolution PDF** is a copy of what's on page 11 and is better for print if you want to print it.
  - We will use this organizer to check-in after you've worked for a few hours.
  - If you can slot in answers to all the areas on the organizer, then you should be able to easily complete the written responses.
  - If **you can't** fill in some area that's an indication of where you should focus your next efforts.
- The **Create PT Completion Timeline** is there as a template for a daily guide for completing the PT.

Review: pages 11 and 12 and add any initial notes or thoughts you have.

- Have students review the suggested timeline and make edits or updates to the schedule that will work for your school situation.
- **Note:** the 12 hours is a minimum. You may opt to give students more time if you wish, but you may not give them less.

Prompt:: Taking into account all the activities we did today plus what you know about the Create PT now, where do you expect you'll be spending most of your time? For which parts of the task should you maximize your time?

Give students a moment to think and share ideas with a partner.

Discuss: Have students share where they think most of their time should go.

- Probably want to maximize writing, video, and code PDF time
- Coding time that isn't focused on making your algorithm or abstraction is likely not well spent. It doesn't matter if your program "looks good" so long as it works!
- Don't forget to allocate time to proofread for easy-to-make mistakes that will cost points, like forgetting to cite sources.

Discussion

Goal: the goal here is to have students start planning in earnest for the Create PT. Students should take seriously how they will allocate their time, and should think about how they probably want to maximize the amount of time they have to write the code and the written responses.

- **Use the response checklists** in the survival guide to make sure you'll earn all the points.

Remarks

Now that we have methods and strategies for completing the task along with the beginnings of a plan, tomorrow we'll start the task in earnest.

Standards Alignment

Computer Science Principles

▸ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.

▸ **7.5** - An investigative process is aided by effective organization and selection of resources. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources.

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes,**contact us.**

Lesson 3: Create PT - Complete the Task (12 hrs)

Project

Overview

It is finally time for students to take on the Create Performance Task. For a total of 12 class hours, students should work on their projects with only types of teacher support allowed (essentially: Advise on process, don't influence or evaluate ideas). Students may also work with a collaborative partner in **in development of their program** - written responses must be done on their own.

The lesson includes reminders about how you can interact with students while they are working on their projects, and suggestions about time line. The Create PT requires a minimum of 12 hours of class time. At the end, students will submit their program code, program video, and written responses through their AP digital portfolio.

Purpose

There are no new CS concepts covered in this lesson. Students will work individually or with a collaborative partner on the Create Performance Task.

Agenda

Getting started

- **Ready?**
- **Read page 3: Preparing for the Through-Course Performance Tasks**
- **Read page 12: Preparing for the Create Performance Task**
- **Read Page 2 - Policy on Plagiarism & Peer-to-Peer Collaboration**
- **Read Pages 12-13 - Guidelines for Completing the Create Performance Task**
- **How to acknowledge others' work, previous work, and cite sources for the Create PT**
- **Set**

Activity (12 hours)

- **GO! Complete Create Performance Task**

Wrap-up

- **Students submit completed Create Performance Task**

View on Code Studio

Objectives

Students will be able to:

- Complete and submit the Create Performance Task.

Preparation

- [ ] Review the **AP CS Principles Course and Exam Description** to understand the teacher's role on the Create PT
- [ ] Review how to create a stand-alone App Lab project to assist students

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teacher

- **AP Digital Portfolio - Teacher Guide**
- **AP CS Principles Course and Exam Description**

For the Students

- **Create PT - Scoring Guidelines 2019** - College Board Doc
- **Create PT - Survival Guide 2018/2019** **Make a Copy**
- **AP Digital Portfolio Access** - College Board Site
- **AP CSP Performance Task Directions for Students** - Resource
- **Code.org Create PT Template** **Make a Copy**
- **Create PT Programming Journal** **Make a Copy**

Teaching Guide

Getting started

Ready?

Remarks

> - For the next ~12 days, you will be completing the Create Performance Task.
> - Most of the work for this task **must** be done independently - but you are allowed to collaborate with another student to create a program. But you must have enough individual contributions to meet all the requirements of the written responses.
> - I (teacher) can help you with **process** and timeline and keep you on task
> - There are a few other last minute things we should look at to make sure you're clear before we start.

Before we start let's look at a few pages in **AP CSP Performance Task Directions for Students - Resource**.

Do these things:

Tech Tips

Creating an App Lab Project: Before students begin programming, they should make sure that they are creating their program in a new App Lab Project - NOT in a level associated with a previous lesson.

However, you can use a previous project as a starting point as long as you add new parts that fulfill the task requirements.

To build off a program started in a previous lesson:

Make sure to click “Remix” from the original program level. This will create a new copy of the program that can be accessed from the list of individual student projects found at **https://studio.code.org/projects**. Students should indicate using comments what parts of the program are copied from older projects.

For students who are creating an entirely new program:

Create a new App Lab project by visiting **https://studio.code.org/projects/applab/new**

How Can You Help as a Teacher: Review the **AP CS Principles Course and Exam Description**, in particular pages 79 - 83 to understand how you as a teacher can and cannot assist on the Create Performance Task.

Read page 3: Preparing for the Through-Course Performance Tasks

This is a general checklist of things you should do to prepare for any of the AP Performance Tasks. Let's see how we did:

Prompt: With a partner - one person reading from the top down, the other reading from bottom up - check off things we've done to prepare so far. Identify anything we **haven't** done, and be sure to note and **Create PT-specific** items.

Quick chat with your partner to summarize anything new or unexpected you discovered.

- Some of the items do not pertain to the Create PT so can be ignored.
- Everything else should have been covered in one way or another if you've done the prep lessons prior to this and used the Create PT Survival guide.
- Address any questions students have

Read page 12: Preparing for the Create Performance Task

This is another checklist of things you should do to prepare specifically for the Create Performance task. Again, let's see how we did.

Prompt: again divide and conquer - with a partner scan the page and check off things you feel confident about, and pull out things you're not sure about.

- Again most of the things on this list are things we've done in the previous preparation lessons.
- They suggest keeping a **journal** to log things like decision points about opportunities and difficulties, algorithm and abstraction -- we recommend using the **Create PT Planning Organizer** as a structured way to log that important information.

Read Page 2 - Policy on Plagiarism & Peer-to-Peer Collaboration

Read the two sections on Page 2, or point out the relevant portions (highlighted below). This is info to have in the background before you go read the guidelines for completing the task next.

Point out Plagiarism applies the the Create PT as well. Specifically:

AP Computer Science Principles Policy on Plagiarism

A student who fails to acknowledge (i.e., through citation, through attribution, by reference, and/or through acknowledgment in a bibliographic entry) the source or author of any and all information or evidence taken from the work of someone else will receive a score of 0 on that performance task.

To the best of their ability, teachers will ensure that students understand ethical use and acknowledgment of the ideas and work of others as well as the consequences of plagiarism. The student's individual voice should be clearly evident, and the ideas of others must be acknowledged, attributed, and/or cited. A computational artifact without acknowledgment of the media used in the creation of the computational artifact, and program code segment(s) written by someone else used in a program without appropriate acknowledgment, are all considered plagiarized work.

Point out Collaboration is allowed as well:

Peer-to-Peer Collaboration

Collaboration is only allowed on designated sub-components of the Create performance task.

For the Explore – Impact of Computing Innovations performance task, collaboration of any kind is not allowed.

For the Create – Applications from Ideas performance task, you are encouraged to collaborate on the development of their program with another student in your class. Collaboration is not allowed during the creation of the video or when answering the written responses.

Students completing AP Computer Science Principles in a nontraditional classroom situation (e.g., online, homeschool, independent study) are encouraged to collaborate with another student peer when completing the Create performance task.

...but you need to be careful that you have enough original work to submit as your own PT.

Hold off discussing these things until after you read the guidelines on pages 12-13.

Read Pages 12-13 - Guidelines for Completing the Create Performance Task

This is a final list of Do's and Don'ts for the Create PT.

Prompt: with your partner, read the **You must**, **You may**, and **You may not** sections of this page. Then with you partner summarize: **what kinds of things can your teacher help you with?**

- Let students read and discuss with partners
- Discuss specifically how you (teacher) are allowed to help and not. Short version: you **can** help students with the **process** of completing the task, you **cannot** help by evaluating their work or ideas in any way.
- There may be some **gray area** around these items:

> - 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

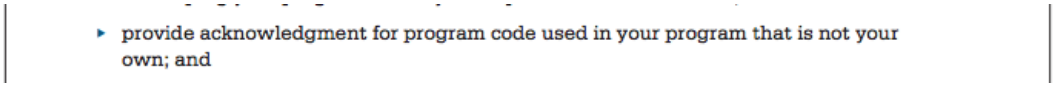
- Taken together this means you **can** submit a project that you had worked on previously, as long as the algorithms and abstractions identified in your response are new and original - and you haven't received feedback on them from a teacher.
- If you modify an existing project for the Create PT **make sure** that the purpose is also new, or modified to fit the changes and updates you are making. For example: "The purpose was to add a login feature to a game I made

previously".

Plagiarism and the Create PT

Students may also notice this line from the guidelines:

You must: ...



How to acknowledge others' work, previous work, and cite sources for the Create PT

- add comments in the program code you submit to indicate which parts were written individually
- In response 2b (development process) make sure to indicate that the work was done collaboratively and refer the reader to comments left in the code.
- In responses 2a and 2b, if building off an earlier project, be sure to mention that the purpose was to extend previous work - whether your own or someone else's - and also make that clear in the development process section.
- If using images, media or other copyrighted material found on the web, you should cite those sources in comments in your program code - usually at the top. Something like:

```
// The images used in this app came from:
// [1] bird image - http://name-of-site.com/path/to/image.jpg
// [2] flower image - http://site.com/path/to/flower.jpg
```

Set

- Take out your Create PT timeline that we developed and reviewed.
- Ask and answer any remaining questions
- Remind students of the overall timeline and that the official PT time is about to start

Activity (12 hours)

GO! Complete Create Performance Task

Remarks

- While completing the performance task, you should keep track of the progress you are making.
- To do this, at the end of each class period record your work in the **Create PT Programming Journal**.
- When finished with your program, use the information in your journal to help complete the Written Response .

Links in Code Studio:

Students can find links to AP documents on the student page in code studio associated with this lesson.

Teaching Tip

- The **Create PT Programming Journal** is designed to be modified to best suit your classroom needs.
- For example, a class with block periods might need less days to complete the hour requirements for the Performance Task. As a result, there could be less total journal entries, but student responses per entry could be longer.
- **Note** It is acceptable to grade these journals for participation. However, it's not appropriate to grade the content of the journals as that could be considered feedback on the student project.

Students may use the **Code.org Create PT Template** to record their Written Response. Remind students that this must be exported as a PDF before uploading to the Digital Portfolio.

Code Studio levels

Create PT - Complete the Task - Student Links

1 *(click tabs to see student view)*

Create PT - Complete the Task

2 *(click tabs to see student view)*

Wrap-up

Students submit completed Create Performance Task

Submitting:

- You are encouraged to submit and save work in the AP digital portfolio as you go!
- At the designated end of the Task administration (having allowed for at least 12 hours of class time for work) students should submit their video of the program running, written responses, and program code to their **AP Digital Portfolio Access - College Board Site**. You can find more instructions as well by using the **AP Digital Portfolio - Teacher Guide**.

Teaching Tip

Submission Timeline: You may spread out submission over a few days if you like since students can save progress in the AP Digital Portfolio. As long as they finalize submission by the closing date of the PTs it's fine.

In the past submitting everything right at the deadline has been a risky proposition as the site sometimes experiences outages due to heavy traffic. Get **something** in early and modify later.

Before they submit their final work:

- Encourage students to check over the **Create PT - Survival Guide 2018/2019** checklists one more time to make sure they met the requirements.
- Make sure they have all the components necessary for the Create Performance Task.

Standards Alignment

Computer Science Principles

- **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- **2.2** - Multiple levels of abstraction are used to write programs or create other computational artifacts
- **4.1** - Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
- **5.1** - Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
- **5.2** - People write programs to execute algorithms.
- **5.3** - Programming is facilitated by appropriate abstractions.
- **5.4** - Programs are developed, maintained, and used by people for different purposes.
- **5.5** - Programming uses mathematical and logical concepts.
- **7.5** - An investigative process is aided by effective organization and selection of resources. Appropriate technologies and tools facilitate the accessing of information and enable the ability to evaluate the credibility of sources.

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes,**contact us.**