Create PT Survival Guide 2021 - 2022

Task Overview

About the Task: During the task you will write a program about a topic entirely of your choosing and then write written responses explaining the purpose, functionality, data abstractions, and procedural abstractions you used. You will have at least 12 class hours to complete the task, at the end of which you'll need to submit:

- A PDF of your program code
- A video of your program running
- Four written responses explaining different parts of your program

About this Guide: By completing the course you should already have the skills and knowledge necessary to complete the task. While the Create Task should be a creative opportunity, **you'll want to make sure you use your time wisely and design your program with the task requirements in mind**. To make sure you're able to do that, this guide includes:

- Activities to highlight tricky parts of the task requirements
- Organizers and planning guides to help you think through ideas
- A recommended schedule for using your 12 hours
- Checklists to make sure your final submission meets all the requirements

College Board Resources: This guide is intended to be a companion to resources provided by the College Board, in particular the task directions and scoring guidelines. Those documents are the final authority on the requirements of the project. This guide, however, should help you understand the nuances of those documents and understand what certain terms mean when designing a program in App Lab.

Getting in the Create PT Mindset: You've been preparing for this moment all year! Make lessons have challenged you to independently decide how to program fully working apps. Unit projects have given you more and more freedom to design projects that take multiple days to program. Throughout the year you've practiced writing written responses and have built key skills in collaboration and debugging. The Create Task will challenge you to put all of these different pieces together. **You're ready for this moment because you've been putting in the practice!**

What is Required of My Program?

You have a lot of freedom to design any project you like for the Create Task. The task directions include only a few requirements for what you include as your program. This page summarizes those requirements, highlights the "takeaways", and explains what they might look like in App Lab.

Input / Output Requirements

The video of your program and response 3a requires you to demonstrate input and output in your program. In App Lab the most straightforward way to do this is to build an app with a user interface. Buttons, dropdowns, or essentially anything that requires you to use an event handler (the onEvent block) is user input. Any time your program plays a sound or shows text or images on the screen that is output.

Takeaway 1: Make sure your program has a user interface with both input and output.

List Requirements

Response 3b requires your program to demonstrate your program uses a list to store multiple pieces of information as well as some code where that list is "processed". This just means you'll need to create a list somewhere in your program, and later you'll want to access the information stored in that list. This list could be a hard-coded list, filled in by user input, or be information you pull from the data library using the `getColumn` block.

Takeaway 2: Make sure your program creates and uses a list of information.

Function Requirements

Your function needs to include "at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration." Sequencing merely means that there are multiple lines of code that run in order. Selection means that a conditional, or if-statement, selects between two or more portions of your program to run. Iteration simply means repeating some behavior, as in a loop.

Takeaway 3: Make sure your program includes a function that has a parameter, an if-statement, and a loop.

Acknowledgment Requirements

The task directions ask that you "Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s)." In theory, you could continue working on a project you started earlier in the year, or you could use code that you found elsewhere. You'll need to be extremely careful in each case that you indicate what is yours and only respond to questions based on the new code you or your partner wrote. You also need to cite any images or sounds that aren't built into App Lab and that you didn't create.

Takeaway 4: Use comments to cite any code, images, sounds, etc. that you or your partner did not create yourselves during the task.

Function Requirement Activity 1 - Does It Count (15 mins)

This and the following activity should help you better understand how to design the function you will submit. Below are some rows from the Rubric for the Create PT.

Criteria	Decision Rules	Scoring Notes
Row 4 The written response: ● includes a program code segment of a student-developed procedure. The procedure: o must have at least one parameter; and o the parameter has an effect on the functionality of the procedure; **AND** ● describes what the selected procedure does and how it contributes to the overall functionality of the program.	The procedure must be student developed, but could be developed collaboratively with a partner. If multiple code segments are included, use the first code segment to determine whether the point is earned. **Do NOT award a point if any one of the following is true:** ● the code segment is an event handler; OR ● the code segment consisting of the procedure is not included in the written responses section; OR ● the written response describes what the procedure does independently without relating it to the overall function of the program.	A procedure is a named group of programming instructions that may have parameters and return values. Procedures are referred to by different names, such as method or function, depending on the programming language. Event handlers are built in abstractions in some languages and will therefore not be considered student-developed. Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.
Row 5 The written response: ● includes a student-developed algorithm that includes: o sequencing, and o selection, and o iteration; **AND** ● explains how the selected algorithm accomplishes its task.	Responses that do not earn row 4 may still earn this row. The algorithm being described can utilize existing language functionality or library calls. An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can get this point. If multiple code segments are included, use the first code segment, as well as any included code for procedures called within this first code segment, to determine whether the point is earned. If this code segment calls other student-developed procedures, the procedures called from within the main procedure can be considered when evaluating whether the elements of sequencing, selection, and iteration are present as long as the code for the called procedures is included. **Do NOT award a point if any one of the following is true:** ● the response only describes what the selected algorithm does without explaining how it does it; OR ● the code segment consisting of the selected algorithm is not included in the written response; OR ● the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm without explicitly identifying the code segment containing the algorithm.).	An algorithm is a finite set of instructions that accomplish a specific task. Every algorithm can be constructed using combinations of sequencing, selection, and iteration. Iteration is a repetitive portion of an algorithm. Iteration repeats until a given condition is met or a specified number of times. The use of recursion is a form of iteration. Selection determines which parts of an algorithm are executed based on a condition being true or false. The use of try/exception statements is a form of selection statements. Program code that is student-developed has been written (individually or collaboratively) by the student who submitted the response. Calls to existing program code or libraries can be included, but are not considered student-developed.

You be the AP Reader! You are are the AP reader trying to determine if they get the point for Rows 4 and 5. Assume each functions below was submitted. For each, select whether Rows 4 and 5 should be awarded and why. No need for detailed explanations, arrows to the code or short bullets are fine.

Example Algorithm 1
 	
`function repeatWord(word, times){`
`  var returnWord = "";`
`  for(var i = 0; i < times; i++){`
`    returnWord = returnWord + word;`
`  }`
`  return returnWord;`
`}`
	Earn Row 4? Yes / No
	
	Earn Row 5? Yes / No
	
	Why?

Example Algorithm 2
 	
`function increaseScore(points){`
 	
`  score = score + 1;`
`  if(score > 10){`
`     endGame();`
`  }`
`}`
	Earn Row 4? Yes / No
	
	Earn Row 5? Yes / No
	
	Why?

Example Algorithm 3
 	
`function addList(list){`
`  var total = 0;`
`  for(var i = 0; i < list.length; i++){`
`    total = total + list[i];`
`  }`
`  return total;`
`}`
	Earn Row 4? Yes / No
	
	Earn Row 5? Yes / No
	
	Why?

Example Algorithm 4
`function addPositives(list){`
`  var total = 0;`
`  var currentItem;`
`  for(var i = 0; i < list.length; i++){`
`    currentItem = list[i];`
`    if(currentItem > 0){`
`      total = total + list[i];`
`    }`
`  }`
`  return total;`
`}`
	Earn Row 4? Yes / No
	
	Earn Row 5? Yes / No
	
	Why?

Function Requirements Activity 2 - Two Function Calls (10 mins)

Response 3d asks you to describe two different calls to the same function with different arguments that cause the function to run differently. For this activity you should:

- Write out two different calls to the function listed on the left with different arguments. For example
    - `findMax([2,3,4])` and `findMax([10,20,30])`
- Describe the specific line of code that will run differently based on the different inputs
- List what the output of each function call will be

```
// This function finds the maximum value in
// a list and returns it.

function findMax(list){
  var max = list[0];
  for(var i = 0; i < list.length; i++){
    if(list[i] > max){
      max = list[i];
    }
  }
  return max;
}
```

Call 1:

Call 2:

Which condition runs differently:

Result of Call 1:

Result of Call 2:

```
// This function checks if the game is over
// If the score is more than 100 then it hides
// all the enemies and runs the endGame function
// that shows your final score.

function checkEndGame(score){
  if(score > 100){
    for(var i = 0; i < 3; i++){
      setProperty("enemy"+i,"hidden",true);
    }
    endGame();
  }
}
```

Call 1:

Call 2:

Which condition runs differently:

Result of Call 1:

Result of Call 2:

Narrow it Down

Why Narrow It Down: You should assume that you're not going to have enough time to complete the "perfect" project for the Create PT. While 12 hours may seem like a long time, the majority of your score actually is based on your written responses, and it turns out that even many small or simple projects meet all the requirements listed above. You'll be better set up for success if you "narrow down" project ideas.

How to Narrow it Down: Narrowing it down means taking a larger idea for your Create Task and finding the smallest version of it that will still meet the task requirements. Here are some ways to do it.

- **Identify your function and list early:** Start out with a clear idea of how you'll hit the minimum requirements of your list and your function.

- **Pick One Part of a Bigger Idea:** Often your original big idea can be broken down to smaller ones that meet the requirements of the task.

- **Minimal Design Mode - looks don't matter:** Complex visual design work in Design Mode (setting colors, fonts, spacing, etc.) will likely NOT meet any of the requirements for the Create PT. Don't worry about how your app looks until after you already have code that will let you complete the written responses.

Practice Narrowing it Down (10 mins)

Below are three descriptions of potential projects that another CS Principles student is considering. For each write:

- Two or three ways they could narrow down the project using the tips above
- Opportunities to write an algorithm in their project even after it's been narrowed down.

Project 1: Tic-Tac-Toe
"Here's my idea: I want to build a tic-tac-toe game. The user creates an account if they don't already have one and are taken to the main game board. From there the player will play against the computer in either easy, intermediate, or advanced mode, so I will need to write the code for the computer player. When the game is over their lifetime win total is updated. I will also keep track of how long the game took."

Ways to narrow down the project (2 or 3)	List and Function opportunities

Project 2: Health App

“I volunteer at my local health clinic so I want to build a health app. The user can record information about what they eat, how much they sleep, how much they exercise, and information like their blood pressure and weight. Based on the information provided the app will provide recommendations to the user about how they can improve their health for both diet and exercise. Users can also personalize the look of the app with different theme colors.”

Ways to narrow down the project (2 or 3)	List and Function opportunities

Project 3: Sports Stats

“I think that I’ll build an app that allows the user to quickly record stats during a basketball game. The app will show a picture of the court. The user taps on the court to indicate something happened there. They are presented with a quick menu of options like: shot attempt, foul, steal, rebound, etc. then they select from another list which player did it. At the end of the game it displays a stat sheet for all of the players and the stats for that game.”

Ways to narrow down the project (2 or 3)	List and Function opportunities

Choosing a Project Idea

You should now understand how the Create PT works and are ready to start brainstorming projects. While you have at least 12 class hours to complete the task, keep in mind that in those 12 hours you must also make your video and complete the written responses. We recommend budgeting at least 5 hours to complete the video and written responses, and so it is highly recommended that you prepare to do a project in which the programming / coding can be completed in 6-7 hours. You want projects with the following features.

- **Personally Relevant:** Pick projects you are actually interested in building.

- **Clear Purpose:** Aim for a simple program whose purpose can be stated in one sentence. For example:
  - The purpose of my program is _______.
  - My program lets a user _______.

- **Narrowed Down:** Repeat the "Narrow it Down" process with your own ideas. A good rule of thumb is that you'll want to be able to have a first draft of your algorithm within two hours of starting to program.

- **No New Programming Skills:** Make sure you *already* have the programming skills necessary to complete the project. Be flexible. With some creativity you can likely use the skills you've already learned to make many different types of projects. Avoid taking on new programming environments or concepts as part of the Create PT.

Brainstorm Project Ideas (10 mins)

Brainstorm one or two project ideas for the Create PT. Afterwards you'll share ideas with a classmate for feedback.

Project Idea	Classmate Feedback
Purpose: *Ways to Narrow it Down:* *List Opportunities:* *Function Opportunities:* *Do I already have the skills to make this project?*	*Use the list above to give feedback on the idea.*
Purpose: *Ways to Narrow it Down:* *List Opportunities:* *Function Opportunities:* *Do I already have the skills to make this project?*	*Use the list above to give feedback on the idea.*

Create PT Written Response Organizer

Complete the final Performance Task using the Create PT Template

3a

Program **purpose:**

3b

Your **list** list name: ________________ type of information: _______________________

Line(s) created: Line(s) processed:	How it manages complexity:

3c

Your **function** that includes a parameter, if-statement, and loop

What it does (think about the experience of the user):	**How** it does it (think about how the code actually works):

3d

Call 1: _______________________________ **Call 2:** _________________________________

Condition that runs differently (include line numbers):

Call 1 Lines that Run and Result:	**Call 2 Lines that Run and Result:**

Create PT Completion Timeline

Before you start, you should think about how you are going to allocate your time for the 12 hours provided for the task. Below is a sample timeline that you can use to plan out how you will complete the Create Performance Task.

Hour	Suggested Activity	Your Plan
1 - 2	Begin building a program for a project you brainstormed. Carefully monitor whether you will finish enough of your project in time. **Goal:** you should be confident after this first round of development that you'll be able to meet the requirements for your list and your function.	
3 - 4	Keep working. Check in after hour 4 once again on whether you are on track to complete responses. You should ideally know: ● The list you will write about ● The function you will write about	
5 - 7	Finalize all programming. After this point you should avoid adding new features and focus on debugging and cleaning up your existing code.	
8	Record **video** of your program running and complete response **3a**	
9	Complete **3b** describing your list and how it is used	
10	Complete **3c** describing your selected function	
11	Complete **3d** describing two different calls to your function	
12	Complete the task ● Review the submission materials ● Check your responses against the scoring guidelines ● Upload your video, written response, and program code to the digital portfolio **Goal:** At the end of this day, your Create PT is submitted!	

Note: The timeline above is just a guideline. You may complete the performance task on a different schedule. Make sure to leave enough time to complete your computational artifact and write-up.

Create PT Guidelines

1. Program Code

Your program must demonstrate:

- output (tactile, audible, visual, or textual) based on input from:
  - the user (including user actions that trigger events); or
  - a device; or
  - a file;
- use of at least one list (or other collection type)to represent a collection of data related to the program's purpose; and
- development of at least one procedure that uses one or more parameters to accomplish the program's intended purpose, and that implements an algorithm that includes sequencing, selection, and iteration.

Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

Create a PDF file that contains all of your program code (including comments).

Advice: For resources on how to make a PDF of your program code head to https://studio.code.org/s/csp8-2021/lessons/4/levels/1. Here's the most important things to remember:

- **Avoiding Plagiarism:** If you fail to cite the source of any code that you or your partner didn't write, you will automatically get a 0 on the entire task. Make sure you use comments to cite any code that you or your partner did not write. For example:

```
// This algorithm for finding the middle value in a list taken from
https://stackoverflow.com/questions/38130895/find-middle-of-a-list/38131003

// 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
```

- **Making Your PDF:** Use CodePrint to make a PDF of your program. It's designed specifically for the Create PT. You can find it from the link above.

2. Video

Your video must demonstrate your program running, including:

- input to your program; and
- at least one aspect of the functionality of your program; and
- output produced by your program.

Your video:

- must be either .mp4, .wmv, .avi, or .mov format; and
- must not exceed 1 minute in length; and
- must not exceed 30MB in file size.

Collaboration is not allowed during the development of your video. Your video must not contain any distinguishing information about yourself. Your video must not be narrated, but text captions are encouraged.

Advice

- **Making Your Video:** Ask your teacher for suggested resources to make a video in your classroom context. Some ideas are provided in the first lesson of this unit as well.
- **Video Runs Continuously:** Your video must run continuously and show your actual code running. It can't just be a series of screenshots.
- **Show One Piece of Functionality:** Your program does NOT need to be complete so long as you can demonstrate one piece of functionality that includes input and output

Video Checklist

❏ Video runs continuously (it cannot be a series of screenshots)
❏ Video is less than 60 seconds long and less than 30MB in size
❏ Video demonstrates input, program functionality, and output

3. Written Responses

Complete the Written Responses using the Create PT Template

3a	**3a.** Provide a written response that: (Approx. 150 words, for all subparts of 3a combined)	**Checklist**
	i. Describes the overall purpose of the program;	❏
	Advice ● **Describe the Purpose:** The purpose of your program is the intended goal or objective of the program. In other words, it's "what" the program is supposed to do. If you made a game, an app, or some other kind of project, just quickly describe "what" kind of program it is and how it would be used / played.	
	ii. Describes what functionality of the program is demonstrated in the video	❏
	Advice ● **Connection to Video:** Make sure that you can connect the purpose of your program to what is shown in the video. If you only have one feature working then describe the purpose of the feature.	
	iii. Describes the input and output of the program demonstrated in the video	❏
	Advice ● **Input and Output:** Make sure you specifically use the words input and output in your response to force yourself to describe specifically how the use provides input in the video and how the program gives back output.	

3b

3b. Capture and paste two program code segments you developed during the administration of this task which contain a list (or other collection type) being used to manage complexity in your program.

Checklist

i. The first program code segment must show how data has been stored in the list. □

Advice

- **Same List Initialized and Processed:** Be very careful that you only include two code segments and that they are both referring to the same list. Even if the processed data is moved to a new list.

ii. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose. □

Advice

- **Same List Initialized and Processed:** Be very careful that you only include two code segments and that they are both referring to the same list. Even if the processed data is moved to a new list

Then, provide a written response that:
(Approx. 200 words, for all subparts of 3b combined, excluding program code)

iii. Identifies the name of the list being processed in this response □

Advice

- **Name Your List:** An easy way to lose points here is simply forgetting to say the name of the list. Start off your response by stating the name of the list.

iv. Identifies what the data contained in the list represent in your program □

Advice

- **Describe The Data:** Explain what type of data is contained in your list and what it stands for in your program and how that helps in fulfilling the program's purpose.

v. Explains how the selected list manages complexity in your program code by explaining how your program code could not be written, or how it would be written differently, if you did not use the list. □

Advice

- **Explaining How It Manages Complexity:** Do NOT simply say that you'd have needed to use variables. Give more details about the specific variables you'd need to create and what information they'd need to store. In some instances it would be nearly impossible to develop your program without a list but be specific in saying how you'd need to change the way it works, again stating specific programming constructs / variables you'd need to use. Explain how your program allows you to use a list of any size without substantial changes to your program code.

3c

3c. Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

Checklist

i. The first program code segment must be a student-developed procedure that:

- Defines the procedure’s name and return type (if necessary)
- Contains and uses one or more parameters that have an effect on the functionality of the procedure
- Implements an algorithm that includes sequencing, selection, and iteration

□

Advice

- **Use this Guide to Pick Your Algorithm:** This Survival Guide has lots of tips for making sure that you have the components necessary to choose an algorithm with the correct components. On a high level you need one function that includes a parameter, a loop, and an if-statement.

ii. The second program code segment must show where your student-developed procedure is being called in your program.

□

Advice

- **Use this Guide to Pick Your Algorithm:** This Survival Guide has lots of tips for making sure that you have the components necessary to choose an algorithm with the correct components. On a high level you need one function that includes a parameter, a loop, and an if-statement.

Then, provide a written response that:
(Approx. 200 words, for all subparts of 3c combined, excluding program code)

iii. Describes in general what the identified procedure does and how it contributes to the overall functionality of the program

□

Advice

- **First the WHAT:** This part asks you to describe what your algorithm does. When answering the "what" question focus on the user's experience (for example: "this function changes the score by one and checks if either player has won" or "this function adds another item to the user's cart after they have selected it and calculates the new total cost".)

iv. Explains in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

□

Advice

- **Next the HOW:** This part asks you to describe how your algorithm does what it does. When answering the "how" question focus on specifically talking through how your code runs, referring to specific lines of code whenever possible (for example: "Lines 20-24 increase the score by adding one to the value in the totalScore variable" or "Lines 110-120 traverse the list storing the prices of all the items in the cart and add them up, storing the total in the totalCost variable")

3d

3d. Provide a written response that:
(Approx. 200 words, for all subparts of 3d combined, excluding program code)

Checklist

i. Describes two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute □

Advice

- **Use this Guide to Pick Your Arguments:** This Survival Guide included an activity where you practiced picking two different arguments for functions.
- **Write Out Both Calls:** Actually include both calls to the function, as in "My first call is updateScore(10) and my second call is updateScore(-10).

ii. Describes what condition(s) is being tested by each call to the procedure □

Advice

- **Use Line Numbers:** This response asks you to name both a specific condition that runs differently and different segments of code that run as a result. In order to do this you need to reference line numbers to clearly demonstrate which code segments are run (or not run) for each condition, as in “The if-statement on line 5 will result in true” and ”Lines 7-10 will run as a result”

iii. Identifies the result of each call. □

Advice

- **Include the Result:** Very clearly state the result for each of the two calls, as in "My first call will return the value 100” and “My second call will return the value -20"