

KEY Create PT Survival Guide 2020 - 2021



Be the AP Reader! You are the AP reader trying to determine if they get the point for Rows 4 and 5. Assume each function 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 <pre>function repeatWord(word, times){ var returnWord = ""; for(var i = 0; i < times; i++){ returnWord = returnWord + word; } return returnWord; }</pre>	Earn Row 4? Yes / No <i>Yes a parameter is included that is used inside the function</i> Earn Row 5? Yes / No <i>No, the function does not include an if-statement (selection)</i> Why?
Example Algorithm 2 <pre>function increaseScore(points){ score = score + 1; if(score > 10){ endGame(); } }</pre>	Earn Row 4? Yes / No <i>No, the parameter is never used in the function</i> Earn Row 5? Yes / No <i>No, the function does not include a loop</i> Why?
Example Algorithm 3 <pre>function addList(list){ var total = 0; for(var i = 0; i < list.length; i++){ total = total + list[i]; } return total; }</pre>	Earn Row 4? Yes / No <i>Yes, the parameter is used inside the function</i> Earn Row 5? Yes / No <i>No, once again an if-statement is not included inside the function (selection)</i> Why?
Example Algorithm 4 <pre>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; }</pre>	Earn Row 4? Yes / No <i>Yes, the parameter is used inside the function</i> Earn Row 5? Yes / No <i>Yes, the function includes examples of sequencing, selection (an if-statement), and iteration (a loop)</i> 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

<pre>// This function finds the maximum value in // a list and returns it. 01 function findMax(list){ 02 var max = list[0]; 03 for(var i = 0; i < list.length; i++){ 04 if(list[i] > max){ 05 max = list[i]; 06 } 07 } 08 return max; 09 }</pre>	Call 1: <i>findMax([30,20,10])</i> Call 2: <i>findMax([10,20,30])</i> Which condition runs differently: <i>Line 04. For call 1 this if statement will never read as true since the largest value is first in the list. For call 2 it will read as true for each element in the list. Each time that means Line 05 will run, updating the max value.</i> Result of Call 1: <i>The maximum value of 30 will be returned</i> Result of Call 2: <i>The maximum value of 30 will be returned</i>
<pre>// 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. 01 function checkEndGame(score){ 02 if(score > 100){ 03 for(var i = 0; i < 3; i++){ 04 setProperty("enemy"+i,"hidden",true); 05 } 06 endGame(); 07 } 08 }</pre>	Call 1: <i>checkEndGame(90)</i> Call 2: <i>checkEndGame(110)</i> Which condition runs differently: <i>The if-statement on Line 02 will evaluate to true for call 2 but not call 1. As a result for call 2 all of lines 3 through 6 will run</i> Result of Call 1: <i>Nothing happens since the score is below 100</i> Result of Call 2: <i>All of the enemies are hidden and the game is ended</i>

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's some ways to do it.

- **Identify you 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
• <i>Build just the login screen</i> • <i>Build an app that can just detect a win/loss/continue condition in tic tac toe</i> • <i>Design a way to keep track of scores over time</i> • <i>Demonstrate a single example of how a computer would choose its next move based on a board setup.</i> • <i>Don't use tic-tac-toe, but a simpler game (pick a number)</i> • <i>Build the user tools that would appear around the game, not the game itself.</i>	<i>List opportunities</i> • <i>The names of people logging into the app</i> • <i>The positions played in tic tac toe</i> • <i>Lists of the win totals or how long each historical game too</i> <i>Function opportunities</i> • <i>Functions to check user names and decide if you can get logged in</i> • <i>Functions to check if someone has won the game of tic tac toe</i> • <i>Functions to update and check the win totals or game length</i>

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)	Algorithm opportunities
• <i>Don't track as much information. Ask for a couple pieces of information and make a single recommendation.</i> • <i>Don't personalize themes or colors</i> • <i>Only provide recommendations for Diet or for Exercise, there's no need to provide recommendations for both.</i> • <i>Focus on the core recommendation portion of the app first. Don't worry about other screens until that piece of code is written.</i>	<i>List opportunities</i> • <i>List of what people eat, when they exercise, their blood pressure, etc.</i> <i>Function opportunities</i> • <i>Any function that processes one of the lists above will likely need to include all the necessary components since it will traverse the list of health data being stored</i>

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)	Algorithm opportunities
• <i>Only track a single statistic at first (e.g. shots, or fouls) as a proof of concept.</i> • <i>Keep the total number of statistics simple (e.g. just a count of shots)</i> • <i>Only keep track of statistics for one team, or even on one side of the court (e.g. offense or defense).</i> • <i>Keep the app to two screens. One for showing shots/fouls/etc. so far and one for showing summary stats.</i> • <i>Avoid adding extra features at first, like an ability to delete statistics added incorrectly.</i>	<i>List opportunities</i> • <i>Any single piece of information this app would track will likely be stored in a list (e.g. shots, attempts, steals, etc.</i> <i>Function opportunities</i> • <i>Any processing of the information above will likely need to be in a function that includes a parameter, a loop, and an if-statement.</i>