Unit 8 Lesson 1

Create PT - Review the Task

Resources

3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video

The purpose of the Word Game Helper app is to assist users in finding words to help in various games like scrabble or crossword puzzles. The video shows how the user selects the length of the word and the first letter from different dropdowns. The user inputs a value to the program using the dropdowns. The output, which is a list of words that meet the chosen conditions, is displayed on the screen.

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program by explaining how your program code would be written differently without using this list

Code Segment #1

```
1  var wordList = getColumn("words", "Word");
```

Code Segment #2

```
15 function filter(len, letter){
16   showElement("waitingImage");
17   filteredWordList = [];
18   setText("output", "");
19
20   for(var i=0; i<wordList.length; i++){
21     if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22         appendItem(filteredWordList, wordList[i]);
23     }
24   }
25
26   if(filteredWordList.length == 0){
27     appendItem(filteredWordList, "No Options Available");
28   }
29
30   hideElement("waitingImage");
31   setText("output", filteredWordList.join(", "));
32 }
```

On line 1, a list called wordList collects all of the words from the "words" database and stores them as a list. Each word is stored as a string. The wordList is used in the program to show users suggested words of various lengths starting with a given letter. For example, if the user selects a length of 2 and the letter "b", the filter function is called in which the wordList is traversed. Every element that is a length of two and start with a "b" is added to a new filtered list. This list is what is outputted to the user.

The wordList manages complexity because without it, each individual word would need to be stored as a single variable, which would then be checked one by one to see if it met the requirements. The program would extend from 32 lines long to thousands of lines long to account for all of these extra variables.

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Code Segment

```
15 function filter(len, letter){
16   showElement("waitingImage");
17   filteredWordList = [];
18   setText("output", "");
19
20   for(var i=0; i<wordList.length; i++){
21     if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22         appendItem(filteredWordList, wordList[i]);
23     }
24   }
25
26   if(filteredWordList.length == 0){
27     appendItem(filteredWordList, "No Options Available");
28   }
29
30   hideElement("waitingImage");
31   setText("output", filteredWordList.join(", "));
32 }
```

The filter function is necessary in order for the program to output a filtered word list to users which meets their chosen requirements. Lines 16 and 30 show and hide an image to let to the user know the program is working. Once the list has been filtered, the image is hidden. To filter the list, a for loop is used (line 20) which traverses wordList. The if statement on Line 21 checks to see if the element at the index is the required length and starts with with the required letter. If it does, the element is added to the filteredWordList (line 22). After the traversal of the list is finished, if the filteredWordList is empty, a string is added to it to let the user know that there are no options available (line 27). Finally, in line 31 the filtered list is displayed to the user, with all the elements joined together with a comma in between each one.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

Let's suppose the user selects from the dropdowns a length of 3 and the letter "d". The arguments passed through the filter function would be 3 and "d" for the parameters *len* and *letter*. In the for loop on line 21, there is an if statement which checks each element in wordList to see if it has a length of 3 and starts with the letter "d". For example, when the element containing "dog" is examined, the conditions (length of 3, first letter "d") is met and therefore the code segment inside of the if statement (line 22) runs and "dog" is added to the filtered list. The for loop continues running checking other elements. After the for loop finishes running, any words that had a length of 3 and started with the letter "d" have been added to the filtered list which is displayed to the user.

Another example would be if the function was called with the arguments 1 and "b". In this case, again the for loop on line 21 is used to traverse the wordList. Each element is checked. When "a" is examined, the conditions (length of 1, first letter "b") is not met. Therefore 21-23 are skipped and the for loop continues on to the next round. Ultimately, no word is found with these conditions, and therefore the filtered list is blank until lines 26-28 where a string is added to let the user know no word was found that met the conditions.

3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video

- Purpose of the program is to help the user decide whether or not to do something
- When the user clicks on the screen, the magic 8 ball appears to make a recommendation and the icons on the screen change to represent if it's a positive, neutral, or negative response.
- Input is the user clicking on the screen. Output is the text displayed on the magic 8 ball and the different icons.

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program by explaining how your program code would be written differently without using this list

Code Segment #1

```
var answers = ["Yes, absolutely", "I have a good feeling about this", "Why not!",
    "Maybe", "I'm not sure", "Ask me again",
    "Don't even think about it!", "Are you kidding?", "You are crazy!"];
```

Code Segment #2

```
// selects a random index
// sets the text of the magic 8 ball to the answer stored at the random index
// play a sound
// calls the function to set the images
function updateScreen(){
  index = randomNumber(0, answers.length-1);
  setText("answerOutput", answers[index]);
  playSound("sound://category_pop/bubble_pop_cluster_2.mp3");
  setImages(index);
}
```

- Name of list = answers
- List of strings which store responses randomly chosen to display on the screen.
- Manages complexity because my code would be longer without a list.

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Code Segment

```
// calls a function to set the images based on the index of the answer
// index {number} - the random index selected when the screen is clicked
function setImages(index){
  if(index < 3){
    styleImages("icon://fa-star", "yellow");
  } else if (index < 6) {
    styleImages("icon://fa-question-circle", "orange");
  } else {
    styleImages("icon://fa-ban", "red");
  }
}

// styles all ten images on the screen
// icon {string} - icon image
// color {string} - icon color
function styleImages(icon, color){
  for(var i=0; i<10; i++){
    setProperty("outputImage" + i, "icon-color", color);
    setProperty("outputImage" + i, "image", icon);
  }
}
```

- The function setImages controls what icons are displayed on the screen after a random response is chosen. This helps the user know if the response was positive or not.
- The function setImages works by choosing an image and calling another function to set the color of the image.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

- Call # 1: setImages(2)
    - The list is organized so the first three elements are positive responses, the next three are neutral, and the last three are negative. This call is passing through the argument 2. Lines 26-27 check if 2 is less than three.
    - This is true, so the response is positive and the styleImages function is run to set the icon to a yellow star.
- Call #2: setImages(7)
    - This call is passing through the argument 7. Lines 26-29 are skipped, because 7 not less than 3 and 7 is not less than 6. Lines 30-32 run because it is the final branch of the if else statement.
    - The response is negative and therefore the styleImages function is run to set the icon to a red crossed out icon.

3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video

The Random Dog Picker app displays a random image and name of a dog based on what size is selected. The video shows several different sizes being chosen and how a different dog shows up each time. The input is the selection in the dropdown, and the output is the image and name of the dog displayed on the screen.

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program by explaining how your program code would be written differently without using this list

Code Segment #1
`3 var dogHeight = getColumn("dogs", "Max Height");`
Code Segment #2

```
21  function filter(){
22    // clears the filtered lists
23    filteredDogNames = [];
24    filteredDogImages = [];
25
26    // gets the size from the dropdown
27    var dogSize = getText("sizeDropdown");
28
29    // traverses the dogHeight List
30    // if dogHeight and dogSize meet certain conditions
31    // the corresponding names and images are stored in the filtered lists
32    for(var i=0; i<dogHeight.length; i++){
33      if(dogHeight[i] < 16 && dogSize == "Small"){
34        appendItem(filteredDogNames, dogNames[i]);
35        appendItem(filteredDogImages, dogImages[i]);
36      } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium")
37        appendItem(filteredDogNames, dogNames[i]);
38        appendItem(filteredDogImages, dogImages[i]);
39      } else if(dogHeight[i] >= 24 && dogSize == "Large") {
40        appendItem(filteredDogNames, dogNames[i]);
41        appendItem(filteredDogImages, dogImages[i]);
42      }
43    }
44
45    // prints the list of dog names that match the value in the dropdown
46    console.log(dogSize + " Dogs:\n" + filteredDogNames);
47  }
```

The list is filtered based on the size selected. If the size chosen is "Small" only dogs whose height is less than 16 will be randomly chosen to be displayed. If the size is "Medium" then the displayed dog needs to be between 16 and 23. If the size is "Large" then dogs that have a height bigger than 24 may be displayed. The list manages complexity because without it, dog heights would all have to be stored in their own individual variables. This would be very confusing, and would add a lot of extra lines.

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Code Segment

```
21  function filter(){
22    // clears the filtered lists
23    filteredDogNames = [];
24    filteredDogImages = [];
25
26    // gets the size from the dropdown
27    var dogSize = getText("sizeDropdown");
28
29    // traverses the dogHeight List
30    // if dogHeight and dogSize meet certain conditions
31    // the corresponding names and images are stored in the filtered lists
32    for(var i=0; i<dogHeight.length; i++){
33      if(dogHeight[i] < 16 && dogSize == "Small"){
34        appendItem(filteredDogNames, dogNames[i]);
35        appendItem(filteredDogImages, dogImages[i]);
36      } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium")
37        appendItem(filteredDogNames, dogNames[i]);
38        appendItem(filteredDogImages, dogImages[i]);
39      } else if(dogHeight[i] >= 24 && dogSize == "Large") {
40        appendItem(filteredDogNames, dogNames[i]);
41        appendItem(filteredDogImages, dogImages[i]);
42      }
43    }
44
45    // prints the list of dog names that match the value in the dropdown
46    console.log(dogSize + " Dogs:\n" + filteredDogNames);
47  }
```

This function filters the lists to smaller lists from which a random name and image can be picked and then displayed. To do this, a for loop in lines 31-43 traverses the dogHeight list and an if else if statement checks to see if each element fits into what the user wants. If it does, the element at that index in the dogName and in the dogImage lists is added to the filtered lists.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

If the user selects a small dog, the function filters the lists to only include small dogs and one of those is chosen randomly and displayed on the screen.
If the user selects a medium dog, the function filters the list to only include medium dogs and of of those is chosen randomly and displayed on the screen.

Create PT 20-21 Code.org Sample 1 - Score: 6/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 1	**1**	**1**	**1**	**1**	**1**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, 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 acknowledgements 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 your program code (including comments).

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.

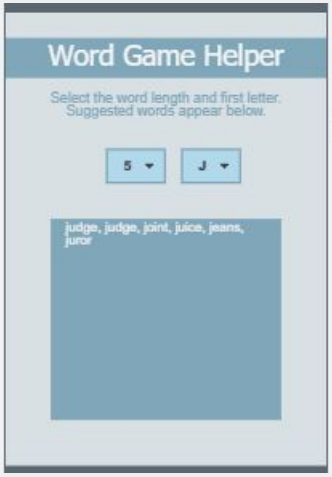
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   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.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

The purpose of the Word Game Helper app is to assist users in finding words to help in various games like scrabble or crossword puzzles. The video shows how the user selects the length of the word and the first letter from different dropdowns. The user inputs a value to the program using the dropdowns. The output, which is a list of words that meet the chosen conditions, is displayed on the screen.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

CRD-2B

● The video demonstrates the running of the program including:
○ input; AND
○ program functionality; AND
○ output
AND
● The response describes the overall purpose of the program.
AND
● The response describes what functionality the video illustrates including input and output shown in the video.</td><td>Do NOT award a point if any of the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The response describes the purpose of the app is "to assist users in finding words to help in various games…" Input and output are shown in the video and described in the response as dropdowns and "list of words...displayed on the screen."</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and

- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="3">

```
1  var wordList = getColumn("words", "Word");
```

```
15 function filter(len, letter){
16   showElement("waitingImage");
17   filteredWordList = [];
18   setText("output", "");
19
20   for(var i=0; i<wordList.length; i++){
21     if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22         appendItem(filteredWordList, wordList[i]);
23     }
24   }
25
26   if(filteredWordList.length == 0){
27     appendItem(filteredWordList, "No Options Available");
28   }
29
30   hideElement("waitingImage");
31   setText("output", filteredWordList.join(", "));
32 }
```

On line 1, a list called wordList collects all of the words from the "words" database and stores them as a list. Each word is stored as a string. The wordList is used in the program to show users suggested words of various lengths starting with a given letter. For example, if the user selects a length of 2 and the letter "b", the filter function is called in which the wordList is traversed. Every element that is a length of two and start with a "b" is added to a new filtered list. This list is what is outputted to the user.

The wordList manages complexity because without it, each individual word would need to be stored as a single variable, which would then be checked one by one to see if it met the requirements. The program
</td><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>

Row 2 - Response 3b

AAP-1.C

The written response:
- includes two program segments:
  - one that shows how the initial data has been stored in this list (or other collection type); and
  - one that shows how the data in this list (or other collection type) is being processed.

AND
- identifies the name of the list being processed in this response

AND
- identifies what the data contained in this list (or other collection type) is representing in the program.

</td><td>The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.

If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.</td></tr>
<tr><td colspan="2">

The response earned the point for this row.
The program code shows how the list is created and processed in the two code segments. The name of the list is identified as wordList. The list stores words that are displayed to the user.

</td></tr>
<tr><td>

Row 3 - Response 3b

AAP-3.C

The written response:
- explains how the named, selected list (or other collection type) manages complexity in the program code by explaining how this program code would be written differently without

</td><td>Responses that do not earn row 2, may still earn this row.

Do NOT award a point if any of the following is true:
- the written response does not name the selected list (or other collection type); OR
- the use of the list is irrelevant or not used in the program; OR
- the explanation does not apply to the selected list; OR
- the explanation of how the list manages complexity</td></tr>
</table>

would extend from 32 lines long to thousands of lines long to account for all of these extra variables.	using this list (or other collection type).	is implausible, inaccurate, or inconsistent with the program; OR ● the solution without the list is implausible, inaccurate, or inconsistent with the program; OR ● the code segments containing the lists are not separately included in the written response section (not including at at all, or the entire program is selected without explicitly identifying the code segments containing the list).
	The response earned the point for this row. The response explains that the code would be written differently without the list by storing each word individually in its own variable, which would require extra program code.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

``
15 function filter(len, letter){
16   showElement("waitingImage");
17   filteredWordList = [];
18   setText("output", "");
19
20   for(var i=0; i<wordList.length; i++){
21   if(wordList[i].length == len && wordList[i].substring(0,1)==letter){
22   appendItem(filteredWordList, wordList[i]);
23   }
24   }
25
26   if(filteredWordList.length == 0){
27   appendItem(filteredWordList, "No Options Available");
28   }
29
30   hideElement("waitingImage");
31   setText("output", filteredWordList.join(", "));
32 }
``
 	
The filter function is necessary in order for the program to output a filtered word list to users which meets their chosen requirements. Lines 16 and 30 show and hide an
	Row 4 - Response 3c
	
	AAP-3.C
	
	The written response:
	● includes a program code segment of a student-developed procedure. The procedure:
	○ must have at least one parameter; and
	○ the parameter has an effect on the functionality of the procedure
	
	AND
		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 response section; OR
		● the written response describes what the procedure does independently without relating it to the overall function of the program.

<table>
<tr><td rowspan="4">image to let to the user know the program is working. Once the list has been filtered, the image is hidden. To filter the list, a for loop is used (line 20) which traverses wordList. The if statement on Line 21 checks to see if the element at the index is the required length and starts with with the required letter. If it does, the element is added to the filteredWordList (line 22). After the traversal of the list is finished, if the filteredWordList is empty, a string is added to it to let the user know that there are no options available (line 27). Finally, in line 31 the filtered list is displayed to the user, with all the elements joined together with a comma in between each one.</td><td>● describes what the selected procedure does an how it contributes to the overall functionality of the program.</td><td></td></tr>
<tr><td colspan="2">The response earned the point for this row.
The code segments is a procedure (function) with parameters (len, letter). The written response explains what the procedure's purpose in the overall program is: "to output a filtered word list to users which meets their chosen requirements."</td></tr>
<tr><td>Row 5 - Response 3c

AAP-2.H, AAP-2,K

The written response:
● includes a student-developed algorithm that includes:
○ sequencing, and
○ selection, and
○ iteration;
AND
● explains how the selected algorithm accomplishes its task.</td><td>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 the 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 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.).</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The code segments displayed an algorithm that included:
● sequencing (more than one line inside the procedure)
● selection (an if-statement)</td></tr>
</table>

	• iteration (a for-loop) The written response explains in detail, line by line, how the algorithm works, which filters a list according to length and the first letter.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">Let's suppose the user selects from the dropdowns a length of 3 and the letter "d". The arguments passed through the filter function would be 3 and "d" for the parameters len and letter. In the for loop on line 21, there is an if statement which checks each element in wordList to see if it has a length of 3 and starts with the letter "d". For example, when the element containing "dog" is examined, the conditions (length of 3, first letter "d") is met and therefore the code segment inside of the if statement (line 22) runs and "dog" is added to the filtered list. The for loop continues running checking other elements. After the for loop finishes running, any words that had a length of 3 and started with the letter "d" have been added to the filtered list which is displayed to the user.

Another example would be if the function was called with the arguments 1 and "b". In this case, again the for loop on line 21 is used</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

CRD-2.J

The written response:
• describes two calls to the selected procedure identified in written response 3c. Each call must pass a different argument causing a different segment of code in the algorithm to execute;
AND
• describes the condition(s) being tested by each call to the procedure;
AND
• identifies the result of each call.</td><td>Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one of the following is true:
• a procedure is not identified in written response 3c or the procedure does not have a parameter; OR
• the two calls cause the same segment of code in the algorithm to execute even if the result is different; OR
• the response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program; OR
• the identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="3">The response earned the point for this row.
The written response clearly explains two different calls to the procedure. Two examples are given with different parameters, which results in different segments of code running. The results of each call are explained "any word that had a length of 3 and started with the letter "d" have been added to the filtered list" and "no word is found with these conditions, and therefore the filtered list is blank until lines 26-28 where a string is added to let the user know no word was found that met the conditions."

Code.org commentary: While the student earned the point, we believe a stronger case for the requirement that "each call must pass a different argument causing a different segment of code to execute" would be made if the code included an if-else or if-else-if statement.</td></tr>
</table>

to traverse the wordList. Each element is checked. When "a" is examined, the conditions (length of 1, first letter "b") is not met. Therefore 21-23 are skipped and the for loop continues on to the next round. Ultimately, no word is found with these conditions, and therefore the filtered list is blank until lines 26-28 where a string is added to let the user know no word was found that met the conditions.	

Create PT 20-21 Code.org Sample 2 - Score: 4/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 2	**1**	**1**	**0**	**1**	**0**	**1**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, 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 acknowledgements 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 your program code (including comments).

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 30 MB 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.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

Purpose of the program is to help the user decide whether or not to do something
When the user clicks on the screen, the magic 8 ball appears to make a recommendation and the icons on the screen change to represent if it's a positive, neutral, or negative response.
Input is the user clicking on the screen. Output is the text displayed on the magic 8 ball and the different icons.
</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

CRD-2B
The video demonstrates the running of the program including:
input; AND
program functionality; AND
output
AND
The response describes the overall purpose of the program.
AND
The response describes what functionality the video illustrates including input and output shown in the video.</td><td>Do NOT award a point if any of the following is true:
the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The student explains that the purpose of the program is "to help the user decide whether or not to do something." The student goes on to explain the functionality the video illustrates: "user clicks on the screen… magic 8 ball appears to make a recommendation… and icons change." The input "user clicking on the screen" and output "text displayed on the magic 8 ball… and the different icons" are also defined in a separate bullet.</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and

- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="4">

```
var answers = ["Yes, absolutely", "I have a good feeling about this", "Why not!",
    "Maybe", "I'm not sure", "Ask me again",
    "Don't even think about it!", "Are you kidding?", "You are crazy!"];
```

```
// selects a random index
// sets the text of the magic 8 ball to the answer stored at the random index
// play a sound
// calls the function to set the images
function updateScreen(){
  index = randomNumber(0, answers.length-1);
  setText("answerOutput", answers[index]);
  playSound("sound://category_pop/bubble_pop_cluster_2.mp3");
  setImages(index);
}
```

- *Name of list = answers*
- *List of strings which store responses randomly chosen to display on the screen.*
- *Manages complexity because my code would be longer without a list.*

</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>

Row 2 - Response 3b

AAP-1.C

The written response:
- includes two program segments:
  - one that shows how the initial data has been stored in this list (or other collection type); and
  - one that shows how the data in this list (or other collection type) is being processed.

AND
- identifies the name of the list being processed in this response

AND
- identifies what the data contained in this list (or other collection type) is representing in the program.

</td><td>

The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment.

If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.

</td></tr>
<tr><td colspan="2">

The response earned the point for this row.
The program code shows how the list is created and processed in the two code segments. The name of the list is identified as answers. The list stores "responses randomly chosen to display on the screen."

</td></tr>
<tr><td>

Row 3 - Response 3b

AAP-3.C

The written response:
- explains how the named, selected list (or other collection type) manages complexity in the program code by explaining how this program code

</td><td>

Responses that do not earn row 2, may still earn this row.

Do NOT award a point if any of the following is true:
- the written response does not name the selected list (or other collection type); OR
- the use of the list is irrelevant or not used in the program; OR
- the explanation does not apply to the selected list; OR

</td></tr>
</table>

	would be written differently without using this list (or other collection type).	• the explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program; OR • the solution without the list is implausible, inaccurate, or inconsistent with the program; OR • the code segments containing the lists are not separately included in the written response section (not including at all, or the entire program is selected without explicitly identifying the code segments containing the list).
	The response DOES NOT earn the point for this row. The response explains that the code would be longer without a list. This is not enough information to earn the point. The students should have explained the specifics of how their code would be different and why the code would be longer.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**

``
// calls a function to set the images based on the index of the answer
// index {number} - the random index selected when the screen is clicked
function setImages(index){
  if(index < 3){
  styleImages("icon://fa-star", "yellow");
  } else if (index < 6) {
  styleImages("icon://fa-question-circle", "orange");
  } else {
  styleImages("icon://fa-ban", "red");
  }
}
 	
// styles all ten images on the screen
// icon {string} - icon image
// color {string} - icon color
function styleImages(icon, color){
  for(var i=0; i<10; i++){
  setProperty("outputImage" + i, "icon-color", color);
  setProperty("outputImage" + i, "image", icon);
  }
}
``
• *The function setImages*
	Row 4 - Response 3c
	
	AAP-3.C
	
	The written response:
	• includes a program code segment of a student-developed procedure. The procedure:
	  ○ must have at least one parameter; and
	  ○ the parameter has an effect on the functionality
		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 response section; OR
		• the written response describes what the procedure does independently without relating it to the overall

controls what icons are displayed on the screen after a random response is chosen. This helps the user know if the response was positive or not. ● *The function setImages works by choosing an image and calling another function to set the color of the image.*	of the procedure AND ● describes what the selected procedure does an how it contributes to the overall functionality of the program.	function of the program.
	The response earned the point for this row. The code segments is a procedure (function) with a parameter (index). The written response explains what the procedure's purpose in the overall program is: "controls what icons are displayed on the screen after a random response is chosen."	
	Row 5 - Response 3c **AAP-2.H, AAP-2,K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing, and ○ selection, and ○ 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 the 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 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.).
	The response DOES NOT earn the point for this row. The code segments displayed an algorithm that included:	

	• sequencing (more than one line inside the procedure) • selection (an if-statement) • iteration (a for-loop) The written response did not explain in detail how the algorithm works. Instead of referring to specific parts of the procedure, the student wrote in general about what the algorithm does instead of HOW it does it.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">• Call # 1: setImages(2)
  ○ The list is organized so the first three elements are positive responses, the next three are neutral, and the last three are negative. This call is passing through the argument 2. Lines 26-27 check if 2 is less than three.
  ○ This is true, so the response is positive and the styleImages function is run to set the icon to a yellow star.
• Call #2: setImages(7)
  ○ This call is passing through the argument 7. Lines 26-29 are skipped, because 7 not less than 3 and 7 is not less than 6.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

CRD-2.J

The written response:
• describes two calls to the selected procedure identified in written response 3c. Each call must pass a different argument causing a different segment of code in the algorithm to execute;
AND
• describes the condition(s) being tested by each call to the procedure;
AND
• identifies the result of each call.</td><td>Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one of the following is true:
• a procedure is not identified in written response 3c or the procedure does not have a parameter; OR
• the two calls cause the same segment of code in the algorithm to execute even if the result is different; OR
• the response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program; OR
• the identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td colspan="3">The response earned the point for this row.
The written response clearly explains two different calls to the procedure. Two examples are given with different parameters, which results in different segments of code running. The results of each call are explained and different lines of code execute depending on the arguments. In the first call, lines 26-27 are executed and "the styleImages function is run to set the icon to a yellow star" whereas with the second call, lines 26-29 are skipped and lines 30-32 run and "the styleImages function is run to set the icon to a red crossed out icon."</td></tr>
</table>

Lines 30-32 run because it is the final branch of the if else statement. ○ *The response is negative and therefore the styleImages function is run to set the icon to a red crossed out icon.*	

Create PT 20-21 Code.org Sample 3 - Score: 2/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	**1**	**0**	**0**	**0**	**1**	**0**

1. **Program Code**
   Your program must demonstrate:
   - output (tactile, 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 acknowledgements 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 your program code (including comments).

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.

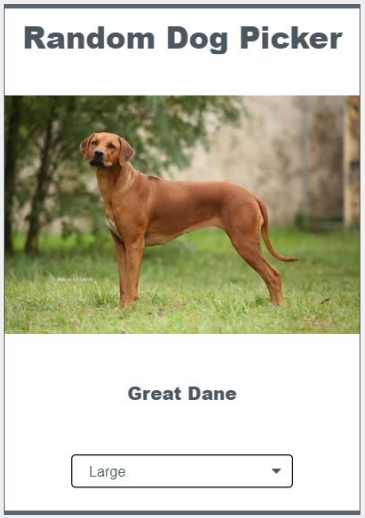
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   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.

3. **Written Responses**
   Submit one PDF file that includes your responses to each prompt below. Clearly label your responses 3a-3d in order. Your responses to all prompts combined must not exceed 750 words, exclusive of the program code. Collaboration is not allowed when answering the written responses.

   3a. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2"> 

The Random Dog Picker app displays a random image and name of a dog based on what size is selected. The video shows several different sizes being chosen and how a different dog shows up each time. The input is the selection in the dropdown, and the output is the image and name of the dog displayed on the screen.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 1
Video and Written Response 3a

CRD-2B

● The video demonstrates the running of the program including:
○ input; AND
○ program functionality; AND
○ output
AND
● The response describes the overall purpose of the program.
AND
● The response describes what functionality the video illustrates including input and output shown in the video.</td><td>Do NOT award a point if any of the following is true:
● the video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited.)</td></tr>
<tr><td colspan="2">The response earned the point for this row.
The student explains that the purpose of the program is to "display[s] a random image and name of a dog based on what size is selected" The student goes on to explain the functionality the video displays: "several different sizes being chosen and… a different dog shows up each time." The input "is the selection in the dropdown" and the output is explained as "the image and name of the dog displayed on the screen."</td></tr>
</table>

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
  *The list is filtered based on the size selected. If the size chosen is "Small" only dogs whose height is less than 16 will be randomly chosen to be displayed. If the size is "Medium" then the displayed dog needs to be between 16 and 23. If the size is "Large" then dogs that have a height bigger than 24 may be displayed. The list manages complexity because without it, dog heights would all have to be stored in their own individual variables. This would be very confusing, and would add a lot of extra lines.*	**Row 2 - Response 3b** **AAP-1.C** The written response: ● includes two program segments: ○ one that shows how the initial data has been stored in this list (or other collection type); and ○ one that shows how the data in this list (or other collection type) is being processed. AND ● identifies the name of the list being processed in this response AND ● identifies what the data contained in this list (or other collection type) is representing in the program.	The written response must include two clearly distinguishable program code segments, but these segments may be disjoint code segments or two parts of a contiguous code segment. If the written response includes more than two code segments, use the first two code segments to determine whether or not the point is earned.
	The response DOES NOT the point for this row. The program code shows how the list is created and processed in the two code segments. However, the name of the list is not identified.	
	Row 3 - Response 3b **AAP-3.C** The written response: ● explains how the named, selected list (or other collection type) manages complexity in the program code by explaining how this program code would be written differently without using this list (or other collection type).	Responses that do not earn row 2, may still earn this row. **Do NOT award a point if any of the following is true:** ● the written response does not name the selected list (or other collection type); OR ● the use of the list is irrelevant or not used in the program; OR ● the explanation does not apply to the selected list; OR ● the explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program; OR

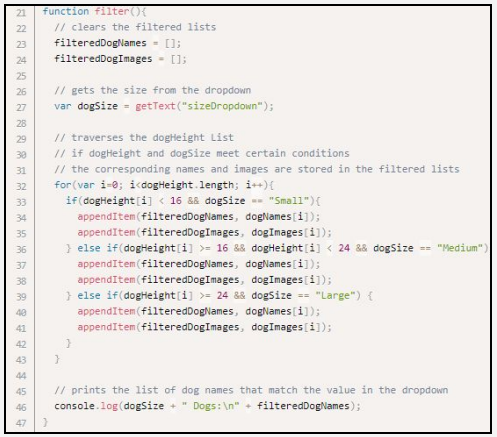
		• the solution without the list is implausible, inaccurate, or inconsistent with the program; OR • the code segments containing the lists are not separately included in the written response section (not including at at all, or the entire program is selected without explicitly identifying the code segments containing the list).
	The response DOES NOT earn the point for this row. According to the decision rules, the student does not earn the point for this row because "the written response does not name the selected list." The student explains how the list manages complexity, but because the list is not specifically named, the answer is invalid.	

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implements an algorithm that includes sequencing, selection, and iteration.

Then, provide a written responses that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

Student Response	**Scoring Guidelines**	
	Row and Task	**Decision Rules**
 *This function filters the lists to smaller lists from which a random*	**Row 4 - Response 3c** **AAP-3.C** The written response: • includes a program code segment of a student-developed procedure. The procedure: ○ must have at least one parameter; and ○ the parameter has an effect on the functionality of the procedure AND • describes what the selected	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 response section; OR • the written response describes what the procedure does independently without relating it to the overall function of the program.

name and image can be picked and then displayed. To do this, a for loop in lines 31-43 traverses the dogHeight list and an if else if statement checks to see if each element fits into what the user wants. If it does, the element at that index in the dogName and in the dogImage lists is added to the filtered lists.	procedure does an how it contributes to the overall functionality of the program.	
	The response DOES NOT earn the point for this row. The code segments is a procedure (function) but the procedure does not use any parameters. The response explains what the procedure does, but does not receive the point because there is no parameter.	
	Row 5 - Response 3c **AAP-2.H, AAP-2,K** The written response: ● includes a student-developed algorithm that includes: ○ sequencing, and ○ selection, and ○ 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 the 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 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.).
	The response earned the point for this row. The code segments displayed an algorithm that included: ● sequencing (more than one line inside the procedure) ● selection (an if-statement) ● iteration (a for-loop) The written response explains how the procedure accomplishes its task: "a for loop in lines 31-43 traverses	

	the dogHeight list and an if else if statement checks to see if each element fits into what the user wants."

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

<table>
<tr><th>Student Response</th><th colspan="2">Scoring Guidelines</th></tr>
<tr><td rowspan="2">If the user selects a small dog, the function filters the lists to only include small dogs and one of those is chosen randomly and displayed on the screen.
If the user selects a medium dog, the function filters the list to only include medium dogs and of of those is chosen randomly and displayed on the screen.</td><th>Row and Task</th><th>Decision Rules</th></tr>
<tr><td>Row 6 - Response 3d

CRD-2.J

The written response:
describes two calls to the selected procedure identified in written response 3c. Each call must pass a different argument causing a different segment of code in the algorithm to execute;
AND
describes the condition(s) being tested by each call to the procedure;
AND
identifies the result of each call.</td><td>Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one of the following is true:
a procedure is not identified in written response 3c or the procedure does not have a parameter; OR
the two calls cause the same segment of code in the algorithm to execute even if the result is different; OR
the response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program; OR
the identified results of either call are implausible, inaccurate, or inconsistent with the program.</td></tr>
<tr><td></td><td colspan="2">The response DID NOT earn the point for this row.
The procedure given in 3c does not have a parameter. Therefore the user is unable to explain two different calls to the function which cause different parts of the code to execute.</td></tr>
</table>

Unit 8 Lesson 2

Create PT - Make a Plan

Resources

Create PT Survival Guide 2020 - 2021

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.

Acknowledgement 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 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.`
 	
`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 }`
	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.`
 	
`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 }`
	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'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

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

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

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 abstraction you will write about ● The algorithm 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 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/csp-create/stage/1/puzzle/2. 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. Provide a written response that:

- describes the overall purpose of the program; and
- describes what functionality the video illustrates; and
- describes the input and output shown in the video.

(Approx. 150 words)

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.
- **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.
- **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.

3a Written Response Checklist:

- ❏ Response describes the overall purpose of the program
- ❏ Response describes the specific functionality shown in the video
- ❏ Response describes both the input and output being shown in the video

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 in your program. The first program code segment must show how data has been stored in the list. The second program code segment must show the data in the same list being processed, such as creating new data from the existing data. Then, provide a written response that:

- identifies the name of the list being processed in this response; and
- identifies what the data contained in the list is representing in your program; and
- explains how the selected list manages complexity in your program code by explaining how your program code would be written differently without using this list.

(Approx. xxx words)

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 and the data it contains to quickly cover the first two bullet points
- **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
- **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.

3b Code Segment Checklist:

- ❏ Only include two program segments: one segment of the list being created and one of the list being processed.
- ❏ The same list is being initialized and processed

3b Written Response Checklist

- ❏ Clearly say the name of the list that is being processed
- ❏ Identify the kind of information being stored in the list
- ❏ State specifically how you would need to rewrite your program by describing the variables you'd need or other ways you'd need to write your program without the list

3c. Capture and paste a procedure from your program that you developed during the administration of this task which implements an algorithm used in your program. This procedure must:

- contain and use one or more parameters that have an effect on the functionality of the procedure; and
- implement an algorithm that includes sequencing, selection and iteration.

Then, provide a written response that:

- describes what the selected procedure does and how it contributes to the overall functionality of the program; and
- explains how the algorithm implemented in the selected procedure accomplishes its task.

(Approx. xxx words)

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.
- **Both WHAT and HOW:** The last two bullets ask you to describe both what your algorithm does, and how it does it. Make sure you answer both questions
    - 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".)
    - 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")

3c Code Segment Checklist:

- ❏ You have copied and pasted the code of a single function
- ❏ You or a partner wrote all the code for this function during the task
- ❏ Your function includes a parameter (sequencing), if-statement (selection), and loop (iteration)

3c Written Response Checklist:

- ❏ Your response explains WHAT your algorithm does in terms of the over program functionality and user experience
- ❏ Your response explains HOW your algorithm works, talking through the different programming constructs you used and referring to specific lines of code.

3d. Provide a written response that:

- describes two calls to the selected procedure identified in written response 3c. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
- describes what condition(s) is being tested by each call to the procedure; and
- identifies the result of each call.

(Approx. xxx words)

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

3d Written Response Checklist

- ❏ Includes two different calls to the function you chose for response 3c
- ❏ You describe the specific condition (if-statement) that will run differently for each call, ideally stating which line of code it appears on
- ❏ You state the different code segments that run for each call, ideally stating the lines of code for each segment
- ❏ You state the result for each call

Unit 8 Lesson 3

Create PT - Complete the Task (12 hrs)

Resources