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.

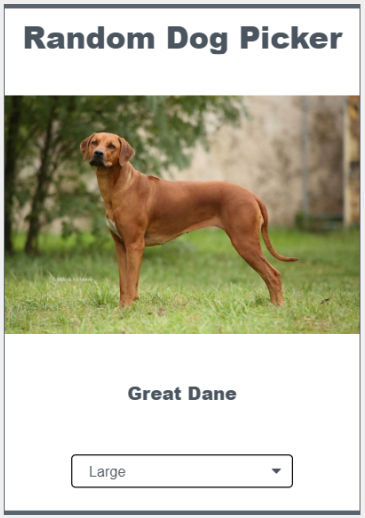
   Your video:
   - must be either .mp4, .wmv, .avi, or .mov format; and
   - must not exceed 1 minute in length; and
   - must not exceed 30 MB in file size.

   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><td>Row and Task</td><td>Decision Rules</td></tr>
<tr><td>Row 1
Video and Written Response 3a

Program Purpose and Function

4.A

CRD-2B

● The video demonstrates the running of the program including:
○ input
○ program functionality
○ output

AND

● The written response:
○ describes the overall purpose of the program.
○ describes what functionality of the program is demonstrated in the video
○ describes the input and output of the program demonstrated in the video.</td><td>Consider ONLY the video and written response 3a when scoring this point.

Do NOT award a point if 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></td><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.

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

```
var dogHeight = getColumn("dogs", "Max Height");
```

```
function filter(){
  // clears the filtered lists
  filteredDogNames = [];
  filteredDogImages = [];

  // gets the size from the dropdown
  var dogSize = getText("sizeDropdown");

  // traverses the dogHeight List
  // if dogHeight and dogSize meet certain conditions
  // the corresponding names and images are stored in the filtered lists
  for(var i=0; i<dogHeight.length; i++){
    if(dogHeight[i] < 16 && dogSize == "Small"){
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 16 && dogHeight[i] < 24 && dogSize == "Medium"){
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    } else if(dogHeight[i] >= 24 && dogSize == "Large") {
      appendItem(filteredDogNames, dogNames[i]);
      appendItem(filteredDogImages, dogImages[i]);
    }
  }

  // prints the list of dog names that match the value in the dropdown
  console.log(dogSize + " Dogs:\n" + filteredDogNames);
}
```

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.

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

Row 2 - Response 3b

Data Abstraction

3.B

AAP-1.C

The written response:

includes two program segments:

one that shows how data has been stored in this list (or other collection type)
one that shows the data in this same list being used as part of fulfilling the program's purpose.

identifies the name of the variable representing the list being used in this response
describes what the data contained in this list is representing in the program.

</td><td>

Consider ONLY written response 3b when scoring this point.

Requirements for program code segments:

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.

Do NOT award a point if the following is true:

The use of the list is trivial and does not assist in fulfilling the program's purpose.

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

Row 3 - Response 3b

Managing Complexity

3.C

</td><td>

Consider ONLY written response 3b when scoring this point.

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

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

	AAP-3.C The written response: ● includes a program code segment that shows a list being used to manage complexity in the program. ● explains how the named, selected list manages complexity in the program code by explaining why the program code could not be written, or how it would be written differently, without using this list.	**Do NOT award a point if any one or more of the following is true:** ● The code segments containing the lists are not separately included in the written response section (not included at all, or the entire program is selected without explicitly identifying the code segments containing the list). ● The written response does not name the selected list (or other collection type). ● The use of the list is irrelevant or not used in the program. ● The explanation does not apply to the selected list. ● The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program. ● The solution without the list is implausible, inaccurate, or inconsistent with the program. ● The use of the list does not result in a program that is easier to develop, meaning alternatives presented are equally complex or potentially easier. ● The use of the list does not result in a program that is easier to maintain, meaning that future changes to the size of the list would cause significant modifications to the code.
	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.

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

```
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.

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

Row 4 - Response 3c

Procedural Abstraction

3.B

AAP-3.C

The written response:

includes two program code segments:

one showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure.
one showing where the student-developed procedure is being called.

describes what the identified procedure does and how it contributes to the overall functionality of the program.

</td><td>

Consider ONLY written response 3c when scoring this point.

Requirements for program code segments:

The procedure must be student developed, but could be developed collaboratively with a partner.
If multiple procedures are included, use the first procedure to determine whether the point is earned.

Do NOT award a point if any one or more 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.

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

Row 5 - Response 3c

Algorithm Implementation

2.B, AAP-2.H, AAP-2.K

The written response:

</td><td>

Consider ONLY written response 3c when scoring this point.

Responses that do not earn row 4 may still earn this row.

Requirements for program code segments:

The algorithm being described can utilize existing

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

	● includes a student-developed algorithm that includes: ○ sequencing ○ selection ○ iteration ● explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it.	language functionality or library calls. ● An algorithm that contains selection and iteration, also contains sequencing. ● An algorithm containing sequencing, selection, and iteration that is not contained in a procedure can earn this point. ● 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 or more of the following is true:** ● The response only describes what the selected algorithm does without explaining how it does it. ● The description of the algorithm does not match the included program code. ● The code segment consisting of the selected algorithm is not included in the written response. ● 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 use of either the selection or the iteration is trivial and does not affect the outcome of the program.
	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

Testing

4.C, CRD-2.J

The written response:
describe two calls to the selected procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.
describes the condition(s) being tested by each call to the procedure.
identifies the result of each call.</td><td>Consider ONLY written response 3d when scoring this point.

Responses that do not earn row 4 may still earn this row.

Do NOT award a point if any one or more of the following is true:
A procedure is not identified in written response 3c or the procedure does not have a parameter.
The written response for 3d does not apply to the procedure in 3c.
The two calls cause the same segment of code in the algorithm to execute even if the result is different.
The response describes conditions being tested that are implausible, inaccurate, or inconsistent with the program.
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>