|  |  |  |
| --- | --- | --- |
|  | **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).

1. **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.

1. **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.

|  |  |  |
| --- | --- | --- |
| **Student Response** | **Scoring Guidelines** | |
| *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.* | **Row and Task** | **Decision Rules** |
| **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. | **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.) |
| **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." | |

**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** | |
| *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 it allows any number of words to be stored in the list instead of using individual variables for each word, 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. If words are added or removed to the dataset this list pulls from, nothing will need to change about the code. The list allows the program to work for any number of words since the filter function will traverse the entire list of words no matter its length.* | **Row and Task** | **Decision Rules** |
| **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. | **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. |
| **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. | |
| **Row 3 - Response 3b**  **Managing Complexity**  **3.C**  **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. | **Consider ONLY written response 3b when scoring this point.**  **Responses that do not earn row 2, may still earn this row.**  **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 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. It also explains that the code allows the program to work for any length word list without any changes. | |

**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** | |
| *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.* | **Row and Task** | **Decision Rules** |
| **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. | **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. |
| **The response earned the point for this row.**  The code segment 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." | |
| **Row 5 - Response 3c**  **Algorithm Implementation**  **2.B, AAP-2.H, AAP-2.K**  The written response:   * 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. | **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 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 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.

|  |  |  |
| --- | --- | --- |
| **Student Response** | **Scoring Guidelines** | |
| *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.* | **Row and Task** | **Decision Rules** |
| **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. | **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. |
| **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. | |