Create PT 20-21 College Board Sample G - Score: 2/6

Total score	Row 1	Row 2	Row 3	Row 4	Row 5	Row 6
Sample: 3	0	1	1	0	0	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
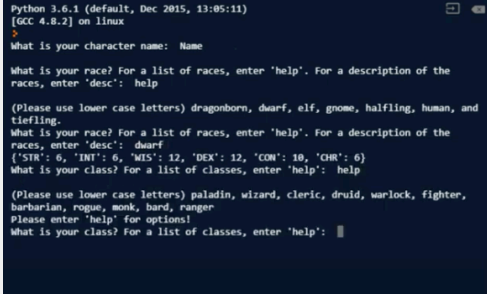   - must not exceed 30 MB in file size.

   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.

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

The program my partner and I wrote addresses the issue of ==making an RPG character==*. Though there are many RPG character creators out there, they can be complicated and difficult to use.* ==Our program is very straightforward with lots of room for customizability in order to allow the easy creation of an RPG character for either a beginner or a seasoned veteran==*.* | **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 DOES NOT the point for this row. The response met only four of the six criteria.**
● The video demonstrates the running of the program to collect input of answers based on a series of output in the form of displayed role-playing game questions and prompts for character creation. This satisfies the first three criteria for the video.
● The response states that the program's purpose is to address "==the issue of making an RPG character==" in a way that is "==very straightforward with lots of room for customizability in order to allow the easy creation of an RPG character for either a beginner or a seasoned veteran==."
● The response does not describe the functionality demonstrated in the video.
● The response does not describe the input and output of the program demonstrated 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.

Student Response	Scoring Guidelines	
	Row and Task	**Decision Rules**
``
3b. stats = {"STR": 0, "INT": 0, "WIS": 0, "DEX": 0, "CON": 0, "CHR": 0}

3b. def StatRoll (index):
    roll1 = (randrange(1, 6)) #rolls
    random value 1-6 roll2 =
    (randrange(1, 6))
    roll3 = (randrange(1, 6))
    value = int(roll1) + int(roll2) + int(roll3) #adds the
    values together stats[index] = value #adds the new value
    to its proper index in stats

Output
3b. print(stats)
```   Dictionary   Changing Values

*The data in the list "stats"* ==represents the stats of the player's character, those being Strength, Intelligence, Wisdom, Dexterity, Constitution, and Charisma. This list manages complexity by allowing all of the stats of the character to be in one area, so they can be easily changed when bonuses arise, certain armor or magic alters them, etc. Without this list, each stat would be its own separate variable. Furthermore, this dictionary will make it much easier to export everything to a document, as all the player stats are in one central location, and it'll take much less formatting==. | **Row 2 - Response 3b**<br><br>**Data Abstraction**<br><br>**3.B, AAP-1.C**<br><br>The written response:<br>- includes two program segments:<br>  - one that shows how data has been stored in this list (or other collection type)<br>  - one that shows the data in this same list being used as part of fulfilling the program's purpose.<br><br>- identifies the name of the variable representing the list being used in this response<br><br>- describes what the data contained in this list is representing in the program. | **Consider ONLY written response 3b when scoring this point.**<br><br>**Requirements for program code segments:**<br>- 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.<br>- 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.<br><br>**Do NOT award a point if the following is true:**<br>- 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, meeting all three criteria.**<br>- The response provides program code segments showing the list stats being initialized and being used in the StatRoll procedure.<br>- The response provides that the list is named ==stats==.<br>- The response describes the data in the list as "==the stats of the player's character, those being Strength, Intelligence, Wisdom, Dexterity, Constitution, and Charisma==." | |
| | **Row 3 - Response 3b** | **Consider ONLY written response 3b when scoring this** |

| | | point. |
|---|---|---|
| | **Managing Complexity**<br><br>**3.C, AAP-3.C**<br><br>The written response:<br><ul><li>includes a program code segment that shows a list being used to manage complexity in the program.</li><li>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.</li></ul> | **Responses that do not earn row 2, may still earn this row.**<br><br>**Do NOT award a point if any one or more of the following is true:**<br><ul><li>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).</li><li>The written response does not name the selected list (or other collection type).</li><li>The use of the list is irrelevant or not used in the program.</li><li>The explanation does not apply to the selected list.</li><li>The explanation of how the list manages complexity is implausible, inaccurate, or inconsistent with the program.</li><li>The solution without the list is implausible, inaccurate, or inconsistent with the program.</li><li>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.</li><li>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.</li></ul> |

**The response earned the point for this row, meeting both criteria.**
- The response provides a list, stats, which is being used to manage complexity in the program.
- The response explains how stats manages complexity in the program "by allowing all of the stats of the character to be in one area, so they can be easily changed when bonuses arise, certain armor or magic alters them, etc. Without this list, each stat would be its own separate variable. Furthermore, this dictionary will make it much easier to export everything to a document, as all the player stats are in one central location, and it'll take much less formatting."

**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 |
|---|---|---|
| ```
looper1 = 1
race = race.lower()
while looper1 == 1:
    if race == "dragonborn": PostRollValue("STR", 2)
    PostRollValue("CHR", 1) print()
    looper1 = 2
    elif race == "dwarf": PostRollValue("CON", 2)
    looper1 = 2
    elif race == "elf": PostRollValue("DEX", 2)
    looper1 = 2
    elif race == "gnome": PostRollValue("INT", 2)
    looper1 = 2
    elif race == "halfling": PostRollValue("DEX", 2)
    looper1 = 2
    elif race == "human": PostRollValue("STR", 1)
    PostRollValue("INT", 1)
    PostRollValue("WIS", 1)
    PostRollValue("DEX", 1)
    PostRollValue("CON", 1)
    PostRollValue("CHR", 1)
    looper1 = 2
    elif race == "tiefling": PostRollValue("INT", 1)
    PostRollValue("CHR", 2)
    looper1 = 2
```

An algorithm that we have developed that includes sequencing, selection, and iteration is the race selection algorithm. This algorithm loops until a proper race is entered, and using if / elif statements, it selects which race was entered. It then uses sequencing to append values to the stats list based on that race. This allows the character to choose the race they want to play, and get its bonuses. | **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 DOES NOT earn the point for this row. The response does not meet any of the criteria.**
• The program code segment provided in response 3c <u>is not a student-developed procedure</u>. While there is a student-developed procedure with parameter in response 3b, program code or answers provided in one prompt cannot be considered when scoring another prompt. | | |

| **Row 5 - Response 3c** | **Consider ONLY written response 3c when scoring this point.** | |

	Algorithm Implementation	**Responses that do not earn row 4 may still earn this row.**
	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.	**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 DOES NOT the point for this row. The response does not meet either of the criteria.
● The response includes a program code segment of a student-developed algorithm that appears to include sequencing, selection (if…elif statement), and iteration (while loop). Even though this algorithm is not included in a procedure with a parameter, it would still be considered. <u>However, because the loop control variable looper1 is initialized to 1 and will be changed to 2 after the first iteration causing the loop to stop, this loop is trivial and the code segment will function the same regardless.</u>

	● The response generally explains the functionality as the "algorithm loops until a proper race is entered, and using if/elif statements, it selects which race was entered. It then uses sequencing to append values to the stats list based on that race." However, <u>the response does not explain how the algorithm works in enough detail to enable someone to recreate it</u>.

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	
	Row and Task	**Decision Rules**
To test this program, many different test cases were used. We tried entering numbers, as well as differently capitalized words, and in all cases we found we had made the program foolproof. One algorithm we tested was the race selection, in which we misspelled words and entered numbers. However, with our loop and .lower, the program worked fine. We also tried entering things other than lawful, neutral, and chaotic in the alignment selection, but it worked perfectly as well.	**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 DOES NOT the point for this row. ● Since <u>there is not a procedure with parameters identified in written response 3c</u>, this response does not provide two specific test cases to a procedure with different paths and outcomes. The response does not meet any of the three criteria.	