Project 1

| **Category** | **Score** | **Actual Score** | **Explanation** |
| --- | --- | --- | --- |
| **Program Purpose and Function** | 0 | 0 | Video fully demonstrates the program functionality, but does not mention the purpose  Does not state the problem being  solved or creative interest being pursued through the program  Poor written response  Incorrect identification of input and output |
| **Data Abstraction** | 0 | 0 | Two code segments, but they do not show any complexity or function  The use of the list does not assist in fulfilling the program’s purpose |
| **Managing Complexity** | 0 | 0 | A list “animallist” is shown, but it does not make the program less complex  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 |
| **Procedural Abstraction** | 0 | 0 | Demonstrated the procedures but they are predetermined  The written response describes what the procedure does independently without relating it to the  overall function of the program |
| **Algorithm Implementation** | 1 | 0 | Identifies the calls used  Differs from collegeboard grading:  The response includes a minimal description, but it does not explain how the algorithm  works in enough detail that someone else could recreate it |
| **Testing** | 1 | 1 | Calls are identified and the correct call results are labeled  Program works as intended |

Project 2

| **Category** | **Score** | **Actual Score** | **Explanation** |
| --- | --- | --- | --- |
| **Program Purpose and Function** | 1 | 1 | The video and written response both clearly demonstrate the program functionality and the purpose  Clearly shown input and output; words and the poem created |
| **Data Abstraction** | 1 | 1 | Two code segments provided  A list is named and identified along with its contents |
| **Managing Complexity** | 1 | 1 | Student explains the importance of the list and how it would not be inconvenient if it wasn't implemented |
| **Procedural Abstraction** | 1 | 1 | The procedure “createpoems” was clearly implemented by the student  Its function is explained in the written response |
| **Algorithm Implementation** | 1 | 1 | The calls are clearly identified and described within the program “createpoems”  Selection and if statements are included |
| **Testing** | 1 | 1 | The results of the calls are successful  The program yields the output of the poems |

Project 3:

| **Category** | **Score** | **Actual Score** | **Explanation** |
| --- | --- | --- | --- |
| **Program purpose and function** | 0 | 0 | Marvel fighting app, but no defined purpose/ function in the video. I  Input and output are described. In the video |
| **Data abstraction** | 1 | 1 | Two clear code segments  the “variable” identified that is used in the response section is “firstCharacter”  The list is defined as “firstCharacterList” |
| **Managing complexity** | 1 | 1 | Purpose is explained  Details why it manages complexity (contains the procedure and uncomplicates appearance). |
| **Procedural abstraction** | 1 | 1 | Clearly explains 2 parameters used in function for character comparison: the characters chosen from list. Explains where procedure is being called and why the comparing of characters is important for the function |
| **Algorithm implementation** | 1 | 1 | Thoroughly explained algorithm that Incorporates sequencing in multiple functions |
| **Testing** | 0 | 1 | I did not see a clear call, but the arguments were apparently the characters  The two calls were described as testing the if/else statements |

Project 4

| **Category** | **My Score** | **Actual Score** | **Explanation** |
| --- | --- | --- | --- |
| **Program purpose and function** | 1 | 1 | Many demonstrations; shows all possible results  Functionality and purpose explained |
| **Data abstraction** | 0 | 0 | Two clear code segments but images are blurry and code could be explained further (storage of data, variables, etc.).  List is without explanation |
| **Managing complexity** | 1 | 0 | Explain the purpose and necessity of the list  It is implausible to assume that the program has infinite guesses without the list, since there could be a simple solution, I thought it would have met criteria |
| **Procedural abstraction** | 1 | 1 | The student thoroughly describes the “checkanswer” procedure and its relation to carrying out the program purpose |
| **Algorithm implementation** | 1 | 1 | There is clearly sequencing and iteration through the check answer algorithm when the user’s inputs are checked again and again after each guess. |
| **Testing** | 0 | 0 | Although I do see 2 calls, there are not 2 arguments listed within the calls to check that the algorithm is in fact working  This makes it impossible to recognize whether or not the program functions |