Submission 1:

| Category | My Score | Actual Score | Explanation |
| --- | --- | --- | --- |
| Program purpose and function | 0 | 0 | Video fully demonstrates the program functionality, but does not mention the purpose explicitly. |
| Data abstraction | 0 | 0 | Two code segments, but they do not show any complexity or function.  The list animalList is not shown or explained to contribute to the purpose of the program. |
| Managing complexity | 0 | 0 | List is titled animallist, but does not demonstrate managing complexity more easily.  “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 | There is a procedure included called evaluateGuess, but there is no explanation on how it contributes to the general purpose of the program as a whole. |
| Algorithm implementation | 1 | 0 | Identifies the for and if loops inside evaluateGuess as meeting the iteration requirements.  According to college board, they shouldn’t get the point because the algorithm is not explained well enough to recreate. |
| Testing | 1 | 1 | Calls are correctly identified and there are conditions for testing evaluateGuess and animalList algorithms. |

Submission 2:

| Category | My Score | Actual Score | Explanation |
| --- | --- | --- | --- |
| Program purpose and function | 1 | 1 | The video clearly demonstrates the program purpose, to create a poem with user inputs. |
| Data abstraction | 1 | 1 | Two code segments provided, with data shown to be stored in wordList, selectWord, and nounList. Parameters are clearly being utilized, and nouns are stored in list. |
| Managing complexity | 1 | 1 | They explain how their list that stores the nouns is integral to the organization of their poem generation. Without this method of storage, their program would be much messier. |
| Procedural abstraction | 1 | 1 | Procedure createpoems clearly explained, and connected back to main purpose of program. |
| Algorithm implementation | 1 | 1 | The calls are clearly identified and described in createpoems procedure. Poem building incorporating if statements and iteration is described. |
| Testing | 1 | 1 | The results of the calls are successful; poems are outputted. Specific calls articlePoem and noArticlePoem are included, and poem output is used to test procedure. |

Submission 3:

| Category | My Score | Actual Score | Explanation |
| --- | --- | --- | --- |
| Program purpose and function | 0 | 0 | Very cool Marvel fighting app, but I didn’t see an actual purpose/ function defined in the video. In the written response, the input and output are nicely described. |
| Data abstraction | 1 | 1 | There are 2 different code segments, which college board requires. The “variable” identified that is used in the response section called firstCharacter. The list defined is firstCharacterList. |
| Managing complexity | 1 | 1 | firstCharacterList’s purpose is explained, and the reason that it manages complexity is nicely explained (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 explains algorithm. Incorporates sequencing in multiple functions, but mostly emphasizes the findWinner one. |
| Testing | 0 | 1 | I didn’t really see 2 different callings of the character comparison algorithm. Conditions were not described either.  They actually did meet the criteria; the arguments were called Carnage and Venom. The two calls were described as testing if else statements explained before in 3c. |

Submission 4:

| Category | My Score | Actual Score | Explanation |
| --- | --- | --- | --- |
| Program purpose and function | 1 | 1 | Many demonstrations; shows all possible results of octordle game. Purpose, to guess the word, is explained at the beginning. The function of the different colors is also explained. |
| Data abstraction | 1 | 0 | There are 2 code segments, and 2 lists are included called work list and guesses. However, the images are a little blurry and code could be explained further (storage of data, variables, etc.).  According to college board, they did not adequately explain how the data stored in the list was used in the second code segment. |
| Managing complexity | 1 | 0 | They explain why the program could not function without a list to store the guesses of words.  According to college board, it is implausible to assume that the program has infinite guesses without the list, since there could be a simple solution. |
| Procedural abstraction | 1 | 1 | The student thoroughly describes checkanswer procedure and its relation to carrying out the purpose of the program. |
| 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 kind of defeats the purpose of calls for testing reasons. |