Student 1

| Category | Criteria | Student Score (2/6) | CB Score (1/6) | Comment |
| --- | --- | --- | --- | --- |
| Program Purpose and Function | * Describes the overall purpose of the program demonstrated in the video. * Describes what functionality of the program is demonstrated in the video. * Describes the input and output of the program. | 0 | 0 | Inputs, outputs, and overall purpose of the function were not described properly. |
| Data Abstraction | * One program code that shows how data has been stored in this list (or another collection type). * One program code 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. | 0 | 0 | Does not explain how the 2 lists shown fulfill the program's purpose. Not a good explanation. |
| Managing Complexity | * 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. | 0 | 0 | Doesn’t explain anything. |
| Procedural Abstraction | * One program code showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. * One program code 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. | 1 | 0 | Does not show a procedure being called, or explain how said procedure works. |
| Algorithm Implementation | * Includes a program code segment of a student-developed algorithm that includes sequencing, selection, and iteration. * Explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. | 0 | 0 | Not explained well. |
| Testing | * Describes 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. | 1 | 1 | Shows two tests with different inputs and results. |

Student 2

| Category | Criteria | Student Score (6/6) | CB Score (6/6) | Comment |
| --- | --- | --- | --- | --- |
| Program Purpose and Function | * Describes the overall purpose of the program demonstrated in the video. * Describes what functionality of the program is demonstrated in the video. * Describes the input and output of the program. | 1 | 1 | Perfect, has everything required. |
| Data Abstraction | * One program code that shows how data has been stored in this list (or another collection type). * One program code 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. | 1 | 1 | Perfect, has everything required. |
| Managing Complexity | * 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. | 1 | 1 | Perfect, has everything required. |
| Procedural Abstraction | * One program code showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. * One program code 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. | 1 | 1 | Perfect, has everything required. |
| Algorithm Implementation | * Includes a program code segment of a student-developed algorithm that includes sequencing, selection, and iteration. * Explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. | 1 | 1 | Perfect, has everything required. |
| Testing | * Describes 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. | 1 | 1 | Perfect, has everything required. |

Student 3

| Category | Criteria | Student Score (6/6) | CB Score (5/6) | Comment |
| --- | --- | --- | --- | --- |
| Program Purpose and Function | * Describes the overall purpose of the program demonstrated in the video. * Describes what functionality of the program is demonstrated in the video. * Describes the input and output of the program. | 1 | 0 | Perfect, has everything required. Could have explained the inputs and outputs slightly better but overall good explanation which earns the point. |
| Data Abstraction | * One program code that shows how data has been stored in this list (or another collection type). * One program code 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. | 1 | 1 | Shows two code segments and lists. Describes what the list represents. |
| Managing Complexity | * 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. | 1 | 1 | Explains how a list is used to manage complexity. |
| Procedural Abstraction | * One program code showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. * One program code 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. | 1 | 1 | Shows a function with parameters. Basically perfect and does everything. Meets all requirements. |
| Algorithm Implementation | * Includes a program code segment of a student-developed algorithm that includes sequencing, selection, and iteration. * Explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. | 1 | 1 | Uses iteration and is very simple for somebody to be able to recreate it. |
| Testing | * Describes 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. | 1 | 1 | Shows the function being called twice with different inputs and the results. |

Student 4

| Category | Criteria | Student Score (5/6) | CB Score (4/6) | Comment |
| --- | --- | --- | --- | --- |
| Program Purpose and Function | * Describes the overall purpose of the program demonstrated in the video. * Describes what functionality of the program is demonstrated in the video. * Describes the input and output of the program. | 1 | 1 | Explains programs purpose, function, inputs, and outputs. Perfect. |
| Data Abstraction | * One program code that shows how data has been stored in this list (or another collection type). * One program code 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. | 0 | 0 | Does not show lists and therefore does not explain what they represent. |
| Managing Complexity | * 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. | 0 | 0 | Does not use a list to manage complexity. |
| Procedural Abstraction | * One program code showing a student-developed procedure with at least one parameter that has an effect on the functionality of the procedure. * One program code 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. | 1 | 1 | Shows a function and its parameters. Describes what the function does and how it contributes to the program. |
| Algorithm Implementation | * Includes a program code segment of a student-developed algorithm that includes sequencing, selection, and iteration. * Explains in detailed steps how the identified algorithm works in enough detail that someone else could recreate it. | 1 | 1 | Uses iteration and selection. Explains how the program could be recreated differently. |
| Testing | * Describes 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. | 1 | 0 | Shows two calls of the function, the parameters, and the input and results of them. |