**Assessment**

**Outcomes covered 2, 3 and 4**

**Part 1**

**Develop interfaces for Abstract Data Types**

Develop an interface for a Set with the following methods:

**SetADT**

add(string) // Adds a string to the set.

remove(string) // Removes a string from the set.

intersection(SetADT s) // sets this set to the intersection of itself and s.

difference(SetADT s) // sets this set to the difference between itself and s.

isize() // returns the number of objects in the set.

isEmpty() // returns true if size = 0, else false.

Submit your interface design.

**Part 2**

**Implement an Abstract Data Type using an array data structure**

Develop an interface for an integer queue with the following methods:

**QueueADT**

push ( value ) //adds value to list appropriately for a queue

pop ( ) //remove and return value from list

isEmpty() //returns true if the queue is empty

size() //returns number of items in queue

Test your code with the following test plan:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Expected Output** | **Actual Output** | **Comments** |
| Check if queue is empty | Message “Queue is empty” | Message “Queue is empty | Show expected result |
| Add 6 integers to queue | None | None | Shows expected result |
| List queue contents | 6 integers displayed | 6 integers displayed | Shows expected result |
| Display number of items in queue | Message “Number of items in queue” 6 | Message “Number of items in queue” 6 | Shows expected result |
| Remove item from queue | Message “Item removed” *display integer removed* | Message “Value removed” *display integer removed* | Used the word Value instead of item |
| List queue contents | 5 integers displayed with first entered missing | 5 integers displayed with first entered missing | Shows expected result |

Submit a copy of your code and the completed test log.