Table 1: Instantiation of a **BasicCollection**

Operation	Purpose	Object State	Expected Result
Collection c = new BasicCollection()	To create a basic collection with the default constructor.	**size = 0** **empty = true**	A new Collection object with default values for the attributes.
c.size()	To verify instantiatied size.		0
c.isEmpty()	To verify instantiated object is empty.		true

Table 2: Adding objects to an instantiated collection.

Operation	Purpose	Object State	Expected Result
BasicCollection <String> c = new BasicCollection<String>()	To create a basic string collection with the default constructor.	**size = 0** **empty = true**	A new Collection object with default values for the attributes.
c.add(new String("A"))	Add a string to the collection.	**size = 1** **empty = false**	The collection contains the string we've added.
!c.isEmpty()	To verify instantiated object isn't empty.		true
c.contains("A")	Check that our collection contains the string added.		true
c.contains("Missing")	To verify that other objects don't falsely report to be in our collection.		false

Table 3: Ability to remove unique elements from the collection.

Operation	Purpose	Object State	Expected Result
BasicCollection <String> c = new BasicCollection<String>()	To create a basic string collection with the default constructor.	**size = 0** **empty = true**	A new Collection object with default values for the attributes.
c.add(new String("A"))	Add a string to the collection.	**size = 1** **empty = false**	The collection contains the string we've added.
c.add(new String("B"))	Add another string to the collection.	**size = 2** **empty = false**	The collection contains the string we've added.
!c.isEmpty()	To verify instantiated object isn't empty.		true
c.size()	Check that our contains two elements.		2
c.remove("B")	To remove an individual item from the collection.	**size = 1**	
c.size()	To check the updated size of our collection.		1
c.contains("B")	To ensure our item was removed.		false

Table 4: Ability to add multiple items to a **BasicCollection**.

Operation	Purpose	Object State	Expected Result
ArrayList<String> list = new ArrayList<>();	Create a random collection that can be used to add to our collection.		
list.add("A");	Add an element to our collection.		
list.add("B");	Add an element to our collection.		
list.add("C");	Add an element to our collection.		
c = new BasicCollection<>();	Create a basic collection using the default constructor.	**size = 0** **empty = true**	A new collection object.
c.addAll(list);	Add our list to our collection.	**size = 3** **empty = false**	A populated collection object.
c.contains("A");	Make sure our collection contains this element that was in the list.		true
c.contains("B");	Make sure our collection contains this element that was in the list.		true
c.contains("C");	Make sure our collection contains this element that was in the list.		true
c.size()	Make sure our collection has the right sizes.		3

Table 5: Ability to remove all items to a **BasicCollection**.

Operation	Purpose	Object State	Expected Result
ArrayList<String> list = new ArrayList<>();	Create a random collection that can be used to add to our collection.		
list.add("A");	Add an item to our list.		
list.add("B");	Add an item to our list.		
list.add("C");	Add an item to our list.		
c = new BasicCollection<>();	Create a basic collection using the default constructor.	**size = 0** **empty = true**	A new collection object.
c.add("A");	Add an item to our collection.	**size = 1** **empty = false**	Our collection.
c.add("B");	Add an item to our collection.	**size = 2**	Our collection.
c.add("C");	Add an item to our collection.	**size = 3**	Our collection.
c.removeAll(list);	Remove all list elements from our collection.	**size = 0** **empty = true**	Our collection has had all the elements removed leaving it empty.
c.contains("A");	Make sure our collection doesn't have an element in it.		false
c.contains("B");	Make sure our collection doesn't have an element in it.		false
c.contains("C");	Make sure our collection doesn't have an element in it.		false
c.size();	Make sure our collection now has the correct size.		0
c.isEmpty();	Make sure our collection is empty.		true

Table 6: When removing all items from a **BasicCollection** don't remove other elements.

Operation	Purpose	Object State	Expected Result
ArrayList<String> list = new ArrayList<>();	Create a random collection that can be used to add to our collection.		
list.add("A");	Add an item to our list.		
list.add("B");	Add an item to our list.		
list.add("C");	Add an item to our list.		
c = new BasicCollection<>();	Create a basic collection using the default constructor.	**size = 0** **empty = true**	A new collection object.
c.add("A");	Add an element to our collection.	**size = 1** **empty = false**	Our collection with one element.
c.add("D");	Add an element to our collection.	**size = 2**	Our collection with two elements.
c.removeAll(list);	Remove the list items from our collection again.	**size = 1**	Our collection still has one element in it.
c.contains("A");	Check that our element doesn't contain the list element.		false
c.contains("D");	Check that our element still contains the other element.		true
c.size();	Check that our collection has the right size.		1
c.isEmpty();	Check that our element isn't empty.		false
BasicCollection <String> c = new BasicCollection<>()	To create a basic string collection with the default constructor.	**size = 0** **empty = true**	A new Collection object with default values for the attributes.
c.add(new String("A"))	Add a string to the collection.	**size = 1** **empty = false**	The collection contains the string we've added.
!c.isEmpty()	To verify instantiated object isn't empty.		true
c.contains("A")	Check that our collection contains the string added.		true
c.contains("Missing")	To verify that other objects don't falsely report to be in our collection.		false