

13. Output from driver (I used the existing bag tester and my own to test the new class):

Testing the Array-Based Bag:

The initial bag is empty.

isEmpty: returns 1; should be 1 (true)

The bag contains 0 items:

Add 6 items to the bag:

The bag contains 6 items:

one two three four five one

isEmpty: returns 0; should be 0 (false)

getCurrentSize: returns 6; should be 6

Try to add another entry: add("extra") returns 0

contains("three"): returns 1; should be 1 (true)

contains("ten"): returns 0; should be 0 (false)

getFrequencyOf("one"): returns 2 should be 2

remove("one"): returns 1; should be 1 (true)

getFrequencyOf("one"): returns 1 should be 1

remove("one"): returns 1; should be 1 (true)

remove("one"): returns 0; should be 0 (false)

The bag contains 4 items:

five two three four

After clearing the bag, isEmpty: returns 1; should be 1 (true)

Testing the ArrayDynamicBag. Test its features that were in ArrayBag and test its unique features.

The initial bag is empty and has a size of 6

isEmpty: returns 1; should be 1 (true)

The bag contains 0 items:

Add 6 items to the bag:

The bag contains 6 items:

one two three four five one

isEmpty: returns 0; should be 0 (false)

getCurrentSize: returns 6; should be 6

Try to add another entry: add("extra") returns 1

contains("three"): returns 1; should be 1 (true)

contains("ten"): returns 0; should be 0 (false)

getFrequencyOf("one"): returns 2 should be 2

remove("one"): returns 1; should be 1 (true)

getFrequencyOf("one"): returns 1 should be 1
remove("one"): returns 1; should be 1 (true)
remove("one"): returns 0; should be 0 (false)

The bag contains 5 items:
extra two three four five

After clearing the bag, isEmpty: returns 1; should be 1 (true)
Bag should be empty. isEmpty() returns 1 (should be 1).

Adding "Example" to the bag.
The bag contains 1 items:
Example

Bag now has a size of 1 (should be 1).

Adding five elements to bag.
The bag contains 6 items:
Example one two three four five

Bag now has a size of 6 (should be 6).

Adding seven elements to bag, should resize and accept new entries.
The bag contains 13 items:
Example one two three four five A B C D E F G

Bag now has a size of 13 (should be 13).

Removing first set of elements.
The bag contains 8 items:
Example G F E D C A B

Bag now has a size of 8 (should be 8).

Removing second set of elements.
The bag contains 1 items:
Example

Bag now has a size of 1 (should be 1).

Remove last element.
Bag should be empty. isEmpty() returns 1 (should be 1).

Add all elements back to ensure downsizing has worked without any weird issues.
The bag contains 12 items:
one two three four five A B C D E F G

Bag now has a size of 12 (should be 12).

Remove all entries once again!

Bag should be empty. isEmpty() returns 1 (should be 1).

All done!