**Code:**

class TestStringBuffer  
{  
 public static void main(String args[])  
 {  
 StringBuffer stb = new StringBuffer("Hi There");  
 StringBuffer stb2 = new StringBuffer("Hi Java Hi There");  
 System.out.println("Capacity Before trimming: " + stb.capacity());  
 System.out.println("Length Before trimming: " + stb.length());  
 stb.trimToSize();  
 System.out.println("Capacity After trimming, before ensureCapaity(): " + stb.capacity());  
 System.out.println("Length After trimming, before ensureCapacity(): " + stb.length());  
 stb.ensureCapacity(30);  
 System.out.println("Capacity after ensureCapaity(), before setLength(): " + stb.capacity());  
 System.out.println("Length after ensureCapacity(), before setLength(): " + stb.length());  
 int length = stb.length();  
 stb.setLength(32);  
 System.out.println("Capacity after setLength(): " + stb.capacity());  
 System.out.println("Length after setLength(): " + stb.length());  
 stb.setLength(length);  
 System.out.println("Deleting First Hi from stb: " + stb.delete(stb.indexOf("Hi"), stb.indexOf("Hi") + "Hi".length()));  
 int indexOfThere = stb.indexOf("There");  
 System.out.println("Before: " + stb + "\nDeleting char \'T\' from stb: " + stb.deleteCharAt(indexOfThere));  
 stb.setCharAt(indexOfThere, 'T');  
 System.out.println("Adding \'T\' back to stb: " + stb);  
 System.out.println("Replacing There with Java in stb: " + stb.replace(indexOfThere, indexOfThere + "There".length(), "Java"));  
 System.out.println("Inserting Hi at the start of stb: " + stb.insert(0, "Hi"));  
 System.out.println("Appending Hi There to stb" + stb.append("Hi There"));  
 System.out.println("Comparing stb with stb2: " + stb.compareTo(stb2));  
 System.out.println("Substring of the stb with portion after last Hi removed: " + stb.substring(stb.lastIndexOf("Hi", stb.length() - 1)));  
 System.out.println("Reverese of stb2: " + stb2.reverse());  
 System.out.println("Finally:\nstb: "+stb +"\nstb2: " + stb2);  
  
 }  
}

**Output:**

Capacity Before trimming: 24  
Length Before trimming: 8  
Capacity After trimming, before ensureCapaity(): 8  
Length After trimming, before ensureCapacity(): 8  
Capacity after ensureCapaity(), before setLength(): 30  
Length after ensureCapacity(), before setLength(): 8  
Capacity after setLength(): 62  
Length after setLength(): 32  
Deleting First Hi from stb: There  
Before: here  
Deleting char 'T' from stb: here  
Adding 'T' back to stb: Tere  
Replacing There with Java in stb: Java  
Inserting Hi at the start of stb: Hi Java  
Appending Hi There to stbHi JavaHi There  
Comparing stb with stb2: 40  
Substring of the stb with portion after last Hi removed: Hi There  
Reverese of stb2: erehT iH avaJ iH  
Finally:  
stb: Hi JavaHi There  
stb2: erehT iH avaJ iH

**Code:**

import java.util.\*;   
class TestVector   
{   
 public static void main(String args[])   
 {   
 Vector<Integer> vec = new Vector<Integer>();  
 System.out.println("\tCapacity of vec: " + vec.capacity());  
 System.out.println("\tSize of vec: " + vec.size());  
 vec.trimToSize();  
 System.out.println("After trimToSize(): ");  
 System.out.println("\tCapacity of vec: " + vec.capacity());  
 System.out.println("\tSize of vec: " + vec.size());  
 for(int i = 1; i <= 10; i++)  
 {  
 vec.add(i);  
 }  
 System.out.println("After Adding Elements: ");  
 System.out.println("\tCapacity of vec: " + vec.capacity());  
 System.out.println("\tSize of vec: " + vec.size());  
 System.out.println(vec);  
 vec.ensureCapacity(30);  
 System.out.println("After ensureCapacity(30): ");  
 System.out.println("\tCapacity of vec: " + vec.capacity());  
 System.out.println("\tSize of vec: " + vec.size());  
 vec.setSize(15);  
 System.out.println("After setSize(15): ");  
 System.out.println("\tCapacity of vec: " + vec.capacity());  
 System.out.println("\tSize of vec: " + vec.size());  
 for(int i = 0; i <= 9; i++)  
 {  
 vec.set(i, 10 + (i % 2));  
 }  
 System.out.println("Setting all elemnts in the Vector to a different value: ");  
 for(int i = 0; i <= 14; i++)  
 {  
 System.out.print(vec.elementAt(i) + " ");  
 }  
 System.out.printf("\n");  
 System.out.println("First Element of the Vector: " + vec.firstElement());  
 System.out.println("Index of first occurence of 11: "+ vec.indexOf(11));  
 System.out.println("Index of first occurence of 11 after index 4: "+vec.indexOf(11, 4));  
 System.out.println("Last Element of the Vector: " + vec.lastElement());  
 System.out.println("Index of last occurence of 11: "+ vec.lastIndexOf(11));  
 System.out.println("Index of last occurence of 11 before index 8: "+ vec.lastIndexOf(11, 8));  
 System.out.println("Initially: " + vec);  
 vec.removeElement(null);  
 System.out.println("Vector after removing a null element\n" + vec);  
 Vector<Integer> vec1 = new Vector<Integer>();  
 vec1.add(null);  
 vec.removeAll(vec1);  
 System.out.println("Vector after removing all null elements\n" + vec);  
 vec.insertElementAt(11, 5);  
 System.out.println("Vector after inserting 11 at index 5 elements\n" + vec);  
 vec.clear();  
 System.out.println("Vector after vec.clear()\n" + vec);  
 System.out.println("The Vector is Empty: " + vec.isEmpty());  
 System.out.println("The Vector contains 11: " + vec.contains(11));  
 }  
}

**Output:**

Capacity of vec: 10  
 Size of vec: 0  
After trimToSize():   
 Capacity of vec: 0  
 Size of vec: 0  
After Adding Elements:   
 Capacity of vec: 16  
 Size of vec: 10  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
After ensureCapacity(30):   
 Capacity of vec: 32  
 Size of vec: 10  
After setSize(15):   
 Capacity of vec: 32  
 Size of vec: 15  
Setting all elemnts in the Vector to a different value:   
10 11 10 11 10 11 10 11 10 11 null null null null null   
First Element of the Vector: 10  
Index of first occurence of 11: 1  
Index of first occurence of 11 after index 4: 5  
Last Element of the Vector: null  
Index of last occurence of 11: 9  
Index of last occurence of 11 before index 8: 7  
Initially: [10, 11, 10, 11, 10, 11, 10, 11, 10, 11, null, null, null, null, null]  
Vector after removing a null element  
[10, 11, 10, 11, 10, 11, 10, 11, 10, 11, null, null, null, null]  
Vector after removing all null elements  
[10, 11, 10, 11, 10, 11, 10, 11, 10, 11]  
Vector after inserting 11 at index 5 elements  
[10, 11, 10, 11, 10, 11, 11, 10, 11, 10, 11]  
Vector after vec.clear()  
[]  
The Vector is Empty: true  
The Vector contains 11: false