1. **Append**
2. class cs{
3. public static void main(String[] args){
4. StringBuffer sb = new StringBuffer("Hello ");
5. sb.append("Java");
6. System.out.println(sb);
7. }
8. }

**Output:**

****

**2. Insert**

class cs{

    public static void main(String[] args){

        StringBuffer sb = new StringBuffer("Hello ");

        sb.insert(1,"JAVA");

        System.out.println(sb);

    }

}

**Output:**

****

**3. Replace**

class cs{

    public static void main(String[] args){

        StringBuffer sb = new StringBuffer("Hello ");

        sb.replace(1,3,"JAVA");

        System.out.println(sb);

   }

}

**Output:**

****

**4. Delete**

class cs{

    public static void main(String[] args){

        StringBuffer sb = new StringBuffer("Hello ");

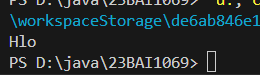
        sb.delete(1,3);

        System.out.println(sb);

    }

}

**Output:**

****

**5. Reverse**

class cs{

    public static void main(String[] args){

        StringBuffer sb = new StringBuffer("Computer");

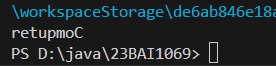
        sb.reverse();

        System.out.println(sb);

    }

}

**Output:**

****

**6. Capacity**

public class capacity {

    public static void main(String[] args) {

        StringBuffer sb = new StringBuffer();

        System.out.println(sb.capacity());

        sb.append("Short note");

        System.out.println(sb.capacity());

        sb.append("Trying medium length");

        System.out.println(sb.capacity());

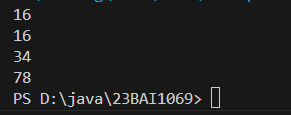
        sb.append("On adding a larger note the capacity is set high");

        System.out.println(sb.capacity());

    }

}

**Output:**

****

**String Builder**

public class stringbuilder {

    public static void main(String[] args) {

        StringBuilder sb = new StringBuilder("Welcome");

        sb.append(" there ");

        System.out.println(sb);

        sb.insert(13,"!");

        System.out.println(sb);

        sb.replace(0,7,"Hello");

        System.out.println(sb);

        sb.delete(0,5);

        System.out.println(sb);

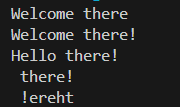
        sb.reverse();

        System.out.println(sb);

    }

}

**Output:**

****

**Capacity**

public class stringbuildercp {

        public static void main(String[] args) {

            StringBuilder sb = new StringBuilder();

            System.out.println(sb.capacity());

            sb.append("Short note");

            System.out.println(sb.capacity());

            sb.append("Trying medium length");

            System.out.println(sb.capacity());

            sb.ensureCapacity(10);

            System.out.println(sb.capacity());

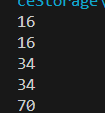
            sb.ensureCapacity(50);

            System.out.println(sb.capacity());

        }

    }

**Output:**

****

**7.Ensure capacity**

public class capacity {

    public static void main(String[] args) {

        StringBuffer sb = new StringBuffer();

        System.out.println(sb.capacity());

        sb.append("Short note");

        System.out.println(sb.capacity());

        sb.append("Trying medium length");

        System.out.println(sb.capacity());

        sb.ensureCapacity(10);

        System.out.println(sb.capacity());

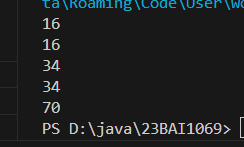
        sb.ensureCapacity(50);

        System.out.println(sb.capacity());

    }

}

**Output:**

****

**Exercise:**

**3rd largest number in array**

import java.util.Scanner;

public class thirdLargestArray {

    public static void main(String[] args) {

        Scanner myobj = new Scanner(System.in);

        System.out.println("Enter length of array: ");

        int len = myobj.nextInt();

        int arr[]= new int[len];

        System.out.println("Enter elements: ");

        for(int i=0;i<len;i++){

            arr[i]= myobj.nextInt();

        }

        for(int j=0;j<len;j++){

           for(int i=j;i<len;i++){

                if(arr[j]<arr[i]){

                    int temp=arr[j];

                    arr[j]=arr[i];

                    arr[i]=temp;

                }

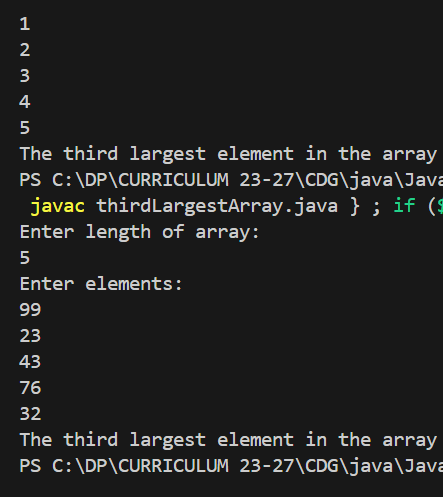
            }

        }

        System.out.println("The third largest element in the array is "+ arr[2]);

    }

}

****

**Output:**