6.1 String Class:

The String class represents character strings. String class is an immutable. The **java.lang.String** class can provide a lot of methods to work on string. By the help of these methods, we can perform operations on string such as copy, trimming, concatenating, converting, comparing, replacing strings etc.
e.g.

String string_name = "abc";

is equivalent to:
char data [] = {'a', 'b', 'c'};
String stringname= new String (data);

Example: illustration of Stringclass:

```
publicclassStringClass
{
        publicstaticvoid main(String [] args)
        {
            String s1 = "abc";
        char data[] = {'p', 'q', 'r'};
            String s2 =new String(data);

            System.out.println("string s1="+s1);
            System.out.println("string s2="+s2);
    }
}
```

Output:
string s1=abc
string s2=pqr

Let us see below list of string methods:

Return type	Method name	description
char	charAt(int index)	Return char at specified index
int	compareTo(String anotherString)	Compare two strings.
int	compareToIgnoreCase (String str)	Compare two strings but ignore case
String	concat(String str)	Connect the specified string to the end of this string
boolean	equals(Object anotherObject)	Compares this string to the specified object
boolean	equalsIgnoreCase(Strin g anotherstring)	Compare this string to another string, ignore case
int	length()	Return the length of this string
String	replace(char oldchar, char newchar)	Return a new string resulting from replacing all occur of old chat in this string with newchar.
String	replace(string reg, String rep)	Replace each substring of this string that matches the given reg with the given rep
String[]	split(string reg)	Split the string if match of the given reg.
String	toLowerCase()	Convert all character in this string to lower case.
String	toUpperrCase()	Convert all character in this string to uppercase.

6.2 Operation on String:

(A) Find out string length:

In java, we can find length of string using length () method. Length method always returns integer value.

Example:

```
publicclass stringoperation
{
        public static void main(String[] args)
        {
         String s1="Welcome";
        int len;
         len=s1.length();
         System.out.println("Length of string is ="+ len);
        }
}
```

Output:

Length of string is =7

(B) Compare two strings:

The equals () method used for compare two string objects. If this method return true then both string object contents are same. We can compare two string objects using compareTo () method compareTo () is quite similar to equals () method.

Example: using equals () method

```
publicclass stringoperation
{
        public staticvoid main(String[] args)
        {
         String s1="Welcome";
         String s2="Welcome";

if(s1.equals(s2))
{
        System.out.println("Both strings are same");
         }
else
{
         System.out.println("Both strings are not same");
}
}
}
```

Output:

Both strings are same

Example 2: using compareTo () method.

```
publicclass stringoperation
{
        publicstaticvoid main(String[] args)
```

```
        {
         String s1="Welcome";
         String s2="Welcome";

if(s1.compareTo(s2)==0)
{
         System.out.println("Both strings are same");
        }
else
{
        System.out.println("Both strings are not same");
}
}
}
```

Output:

Both strings are same

(C) Connect two strings:

Two or more strings putting together is called concatenation.in java concatenation is perform using "+" operator and concat () method.

Example:

```
publicclass stringoperation
{
    publicstaticvoid main(String[] args)
        {
         String s1="Welcome";
        String s2=" to java programming";
         String s3="";
        s3=s1.concat(s2);
        System.out.println("After connect two strings is =" +s3);
        }
}
```

Output:

After connect two strings is =Welcome to java programming

(D) Convert string to upper case:

In java, to change the lower case to upper case using toUpperrCase () method.

Example:

```
publicclass stringoperation
{
        publicstaticvoid main(String[] args)
        {
                String s1="welcome to java programming";
                String s2="";
                s2=s1.toUpperCase();
                System.out.println("After convert upper case  = " +s2);
        }
}
```

Output:
After convert upper case =WELCOME TO JAVA PROGRAMMING

(E) Convert string to lower case:

In java, to change the Upper case to Lower case using toLowerCase () method.

Example:

```
publicclass stringoperation
{
        publicstaticvoid main(String[] args)
        {
                String s1="WELCOME TO JAVA PROGRAMMING";
                String s2="";
                s2=s1.toLowerCase();
                System.out.println("After convert Lower case  = " +s2);
        }
}
```

Output:

After convert Lower case = welcome to java programming

6.3 StringBuffer Class:

In java, StringBuffer class is a mutable (modifiable) means we can modify the string using StringBuffer class methods. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

Let us see below list of StringBuffer Constructor:

Constructor	description
StringBuffer():	Creates an empty string buffer with the initial capacity of 16.
StringBuffer(String str):	Creates a string buffer with the specified string.
StringBuffer(int capacity):	Creates an empty string buffer with the specified capacity as length.

Let us see below list of StringBuffer Method name:

Return Type	Method name	Description
synchronized StringBuffer	append(String s)	This method is used to append the specified string with this string.
synchronized StringBuffer	insert(int offset, String s)	This method is used to insert the specified string with this string at the specified position.
synchronized StringBuffer	replace(int startIndex, int endIndex, String str)	This method is used to replace the string from specified startIndex and endIndex.
synchronized StringBuffer	delete(int startIndex, int endIndex)	This method is used to delete the string from specified startIndex and endIndex.
synchronized StringBuffer	reverse()	This method is used to reverse the string.

int	capacity()	This method is used to return the current capacity.
char	charAt(int index)	This method is used to return the character at the specified position.
int	length()	This method is used to return the length of the string i.e. total number of characters.

Example 1:

```
public class StringBufferClassDemo
{
        public static void main(String [] args)
        {
                StringBuffer sb=new StringBuffer("Hello ");
                sb.append("world");
                System.out.println(sb);                         //print Hello world

                StringBuffer sb2=new StringBuffer("Hello ");
                sb2.insert(1,"world");
                System.out.println(sb2);                        //print  Hworldello

                StringBuffer sb3=new StringBuffer("Hello");
                sb3.replace(1,3,"world");
                System.out.println(sb3);                        // print Hworldlo

                StringBuffer sb4=new StringBuffer("Hello");
                sb4.delete(1,3);
                System.out.println(sb4);                         // print Hlo

                StringBuffer sb5=new StringBuffer("Hello");
                sb5.reverse();
                System.out.println(sb5);                         //print olleH

                StringBuffer sb6=new StringBuffer();
                System.out.println(sb6.capacity());             //print 16

                StringBuffer sb7=new StringBuffer();
                sb7.append("java is good programming language");
                System.out.println(sb7.capacity());             //print 34
                }
}
```

Output:

```
Hello world
Hworldello
Hworldlo
Hlo
olleH
16
34
```