

Understanding String Handling: String accepts a group of characters.

If we want to take String data we need to keep that data inside " "

String Class: Strings which were created by using String Class are **Immutable**

```
String s1="Java"; // Java
```

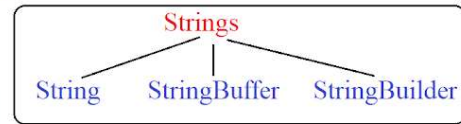
```
String s2=new String("Java"); // Java
```

```
char arr[]={'J','a','v','a'};
```

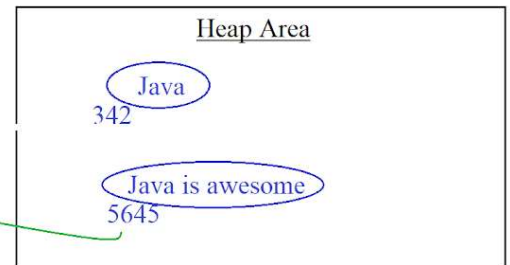
```
String s3=new String(arr); // Java
```

```
String s4=new String(arr,1,2); //??
```

Immutability means once we have created a String class object we will not be able to modify its contents, if we are trying to modify entirely a new object will be getting created old object will not be affected with these changes.



```
String s="Java";  
s=s.concat(" is awesome");
```



Understanding String Handling

- “A String represents group of characters”. Strings are represented as String objects in java.
- After creating a String object, we can't perform any changes to the existing object.
- If we are trying to perform any changes to that string, those changes will be appended to a new object will be created.
- This behavior is called immutability of the String object.
- String objects are **immutable**.

Creating Strings:

- We can declare a String variable and directly store a String literal using assignment operator.

String str = "Hello";

- We can create String object using new operator with some data.

String s1 = new String ("Java");

- We can create a String by using character array also.

char arr[] = { 'p','r','o','g','r','a','m'};

- We can create a String by passing array name to it, as:

String s2 = new String (arr);

- We can create a String by passing array name and specifying which characters we need:

String s3 = new String (arr 2, 3);

- Here starting from 2nd character a total of 3 characters are copied into String s3.

- Merging of two strings is known as 'String Concatenation'.
- We can perform concatenation operation by using '+' operator or by using concat().
- If we are creating a String in the following way

String s=new String("Java");

then two objects will be created one is in the heap memory the other one is String constant pool (SCP).

- The identifier 's' will be pointing towards the object which is present in the heap memory.
- If we are creating a String in the following way

String s="Java";

- Then only one object will be created in SCP and 's' will be always referring that object.

- Object creation in SCP is always optional 1st JVM will check is any object already created with required content or not. If it is already available then it will reuse existing object instead of creating new object.
- If it is not already there then only a new object will be created. Hence there is no chance of existing 2 objects with same content on SCP that is **duplicate objects are not allowed in SCP**.
- Garbage collector can't access SCP area hence even though object doesn't have any reference still that object is not eligible for GC if it is present in SCP.
- All un referenced/null SCP objects will be destroyed at the time of JVM shutdown automatically.
- In **SCP** If an object is already available it will reuse the existing object instead of creating new object.
- In **SCP** there is no chance of duplicate objects.
- When ever we are using new operator compulsory a new object will be created in the Heap Memory.
- There may be a chance of duplicate objects in heap memory but there is no chance duplicate objects in SCP.

- Advantage of SCP:

Instead of creating a separate object for every requirement we can create only one object and we can reuse same object for every requirement. This approach improves performance and memory utilization.

- Disadvantage of SCP:

As several references pointing to the same object, if we change the value of one reference all other references pointing to that object will be reflected with that, in order to prevent this Strings are immutable.

Array is not growable in run time the length is fixed

String is a predefined class.

In Strings we have

1.String Class.

2.String Buffer Class

3.String Builder Class

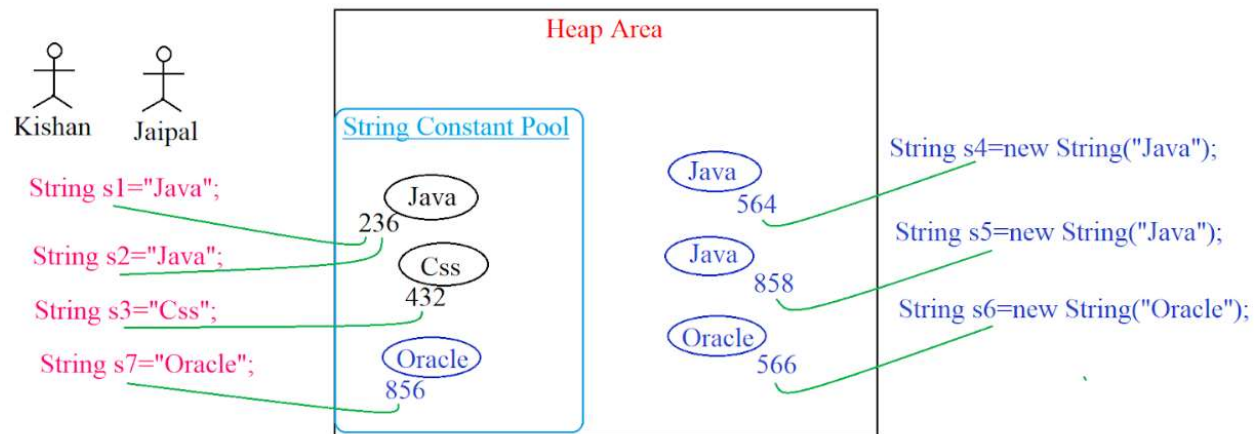
Immutability (not constant) –once we are created a string class object, we will not be able to modify its content, if we

are trying to modify entirely a new object getting created old object will not be effected.

```
3 public class ClassA
4 {
5     void meth1()
6     {
7         System.out.println("Implementing String Handling\n");
8
9         String s1="Java";
10        String s2=new String("Java");
11        char arr[]= {'J','a','v','a'};
12        String s3=new String(arr);
13        String s4=new String(arr,2,1);
14
15        System.out.println("s1 : "+s1+"("+s1.length()+")");
16        System.out.println("s2 : "+s2+"("+s2.length()+")");
17        System.out.println("s3 : "+s3+"("+s3.length()+")");
18        System.out.println("s4 : "+s4+"("+s4.length()+")");
19    }
20    void meth2()
21    {
22        System.out.println("Strings which were created by using String Class are Immutable\n");
23        String s="Java";
24        System.out.println("Before s : "+s);
25
26        //System.out.println(s.concat(" is awesome"));
27        s=s.concat(" is awesome");
28
29        System.out.println("After s : "+s);
30
31    }
32 }
33 public static void main(String[] args)
34 {
35     ClassA aobj=new ClassA();
36     //aobj.meth1();
37     aobj.meth2();
38 }
39 }
```

String s="Java"; \Rightarrow **Case 1**

String s=new String("Java"); \Rightarrow **Case 2**



NOTE: In SCP there is NO chance of Duplicate Objects