Lesson:

Strings in Java

List of Concepts Involved:

- Way to compare
- Inbuilt methods in String class
- Concatenation

Different ways of Comparison

To compare 2 Strings in java we use following approach

a. == operator

It compares the reference of the Object.

b. equals()

It compares the contents of two objects.

Example

```
String s1 = new String("sachin");
String s2 = new String("sachin");
System.out.println(s1==s2); //false
System.out.println(s1.equals(s2));//true
```

Important methods of String

1. public char charAt(int index)

Example

```
String s="sachin";
System.out.print(s.charAt(0));//s
System.out.print(s.charAt(-1));//StringArrayIndexOutOfBoundsException
System.out.print(s.charAt(10));//StringArrayIndexOutOfBoundsException
```

2. public String concat(String str)

Example

```
String s="sachin";
System.out.println(s.concat("tendulkar"));
s+="IND";
=s+"MI";
System.out.print(s);
```

3.public boolean equals(Object o)

It is used for Content Comparison,In String class equals() method is Overridden to check the content of the object.

It is used for Content Comparison without comparing the case.

5. public String subString(int begin)
It gives the String from the beginning index to the end of the String.

Example
```
String s="iNeuron";
System.out.print(s.substring(2));//searching from 2 to end of the string
```

6. public String subString(int begin,int end)
It gives the String from the begin index to end-1 of the String.

Example
```
String s="INeuron";
System.out.print(s.substring(2,6));//searching from 2 to 5 will happen
```

7. public int length()
It returns the no of characters present in the String.

Example
```
String s="INeuron";
System.out.print(s.length());//7
System.out.print(s.length);//Compile time error
```

8. public String replace(char old, char new)

Example
```
String s="ababab";
System.out.print(s.replace('a','b')); //bbbbbb
```

9. public String toLowerCase()
10. public String toUpperCase()
11. public String trim()
To remove the blank spaces present at the beginning and end of string but not the blank spaces present at the middle of the String.

12. public int indexOf(char ch)
It returns the index of 1st occurance of the specified character if the specified character is not available then it returns -1.

Example
```
String s="sachinramesh";
System.out.print(s.indexOf('a'));//1
System.out.print(s.indexOf('z'));//-1
```

13. public int lastIndexOf(char ch)
It returns the index of last occurrence of the specified character if the specified character is not available then it returns -1.

Example
```
String s="sachinramesh";
System.out.print(s.lastIndexOf('a'));//7
System.out.print(s.lastIndexOf('z'));//-1
```

Note:
Because of runtime operation, if there is a change in the content with those changes a new Object will be created only on the heap, but not in SCP.
If there is no change then the same object will be reused.
This rule is applicable for Objects present in both SCP and Heap.

Example1
```
String s1="sachin"
String s2=s1.toUpperCase();
String s3=s1.toLowerCase();
System.out.print(s1==s2);//false
System.out.print(s1==s3);//true
```

Example2
```
String s1="sachin";
String s2=s1.toString();
System.out.print(s1==s2);//true
```

Example3
```
String s1=new String("sachin");
String s2=s1.toString();
String s3=s1.toUpperCase();
String s4=s1.toLowerCase();
String s5=s1.toUpperCase();
String s6=s1.toLowerCase();
System.out.print(s1==s6);//true
System.out.print(s3==s5);//false
```

Concatenation

Concatenation is the process of combining two or more strings into a single string. This can be done in multiple ways, including using the "+" operator or the concat() method.

Example 1:
Using the "+" operator:
String str1 = "Hello";
String str2 = "World";
String str3 = str1 + " " + str2;
System.out.println(str3); //Output: Hello World

Example 2:
Using the concat() method:

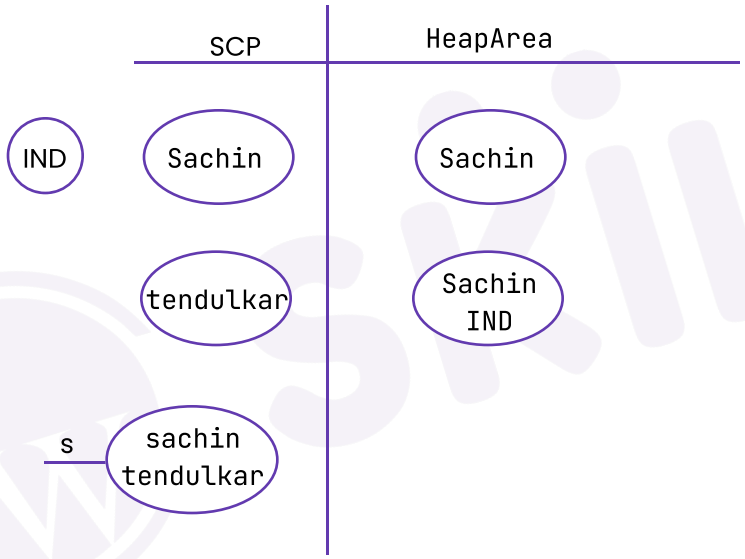
```
String str1 = "Hello";
String str2 = "World";
String str3 = str1.concat(" ").concat(str2);
System.out.println(str3);    //Output: Hello World
```

Example3:

```
String s = new String("sachin");
s.concat("tendulkar");
s=s.concat("IND");
s="sachintendulkar";
```

Output

Direct literals are always placed in SCP. Because of runtime operations, if an object is required to be created compulsorily, that object should be placed on the heap, not in SCP.



Example 4:

```
String s1= new String("sachin");
s1.concat("tendulkar");
s1+="IND";
String s2=s1.concat("MI");
System.out.println(s1);
```