



# String

3 concept

# Difference b/w



```
String s1 = new String("manshu");  
String s2 = "manshu";
```





# String object

```
String s1= new String ("manshu");
```

In this case two objects will be created one is in the heap

other one is in SCP(String constant pool) and

s1 is always pointing to heap object.



# String literal

```
String s2 = "manshu";
```

In this case only one object  
will be created in SCP  
and  
s is always  
referring that object.



**Heap**

**SCP**

**s2 -> manshu**



# most imp difference

---



Whenever we are using new operator compulsory a new object will be created on the Heap .

There may be a chance of existing two objects with same content on the heap

but

there is no chance of existing two objects with same content on SCP .



## conclusion

---



In heap  
Duplicates are  
allowed



in SCP  
Duplicates are  
not allowed



# Goals

```
String s1= new String ("manshu");  
String s2 = new String ("manshu");  
String s3 = new String ("manshu");  
String s4 = new String ("manshu");  
String s5 = new String ("manshu");
```



## Heap

**s1-->manshu**  
**s2-->manshu**  
**s3-->manshu**  
**s4-->manshu**  
**s5-->manshu**

## SCP

**manshu**





# Goals

```
String s1 = new String("shivam");  
String s2 = new String ("shivam");  
String s3 = "shivam";  
String s4 = "shivam";
```

## conclusion



In heap

s1--> shivam  
s2--> shivam



in SCP

shivam  
s3 s4

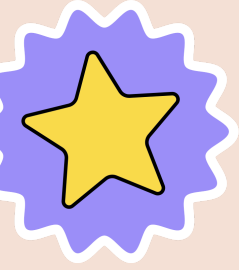
== vs equal()

---

in String class







```
String s1 = new String("manshu");  
String s2 = new String("manshu");  
String s3 = new String("shivam");  
String s4 = "manshu";  
String s5 = "manshu";  
String s6 = "shivam";
```

### Heap

**s1-->manshu**  
**s2-->manshu**  
**s3 --> shivam**

### SCP

**s4-> manshu**  
**s5-> manshu**  
**s6-> shivam**



```
String s1 = new String("manshu");  
String s2 = new String("manshu");
```



`s1 == s2 ?`

no

because both are in heap and  
referring to different object

`s1.equals(s2)`

yes

because both content are same

**Heap**

**s1-->manshu**  
**s2-->manshu**  
**s3 --> shivam**

**SCP**

**s4->**  
**s5-> manshu**  
**s6-> shivam**

```
String s1 = new String("manshu");  
String s4 = "manshu";
```



`s1 == s4 ?`

no

because s1 is in heap pointing to different than that of s4 which is in scp

`s1.equals(s4)`

yes

because both content are same

**Heap**

**s1-->manshu**  
**s2-->manshu**  
**s3 --> shivam**

**SCP**

**s4-> manshu**  
**s5-> manshu**  
**s6-> shivam**

```
String s4 = "manshu";  
String s5 = "manshu";
```

**s4 == s5 ?**

**yes**

because both s4 and s5 are in  
SCP and in SCP duplicate are  
not allowed thus both are  
pointing to same object

**Heap**

**s1-->manshu  
s2-->manshu  
s3 --> shivam**

**SCP**

**s4-> manshu  
s5-> manshu  
s6-> shivam**



# Immutability

---

String can not change

same like

what you have written with  
pen that can't be change





Once we create a String object we can't perform any changes in the existing object.

If we are try to perform any changes  
with those changes  
a new object will be created.

This behavior is called  
immutability of the String object.





For every String Constant one object will be created in SCP.

Because of runtime operation if an object is required to create compulsory that object should be placed on the heap but not SCP.





```
String s=new String("manshu");  
    s.concat(" shivam");  
System.out.println(s);//manshu
```

**Heap**

**s-->manshu**

(no reference)

**manshu shivam**

**SCP**

**manshu**

**shivam**







```
String s= "manshu";  
s.concat(" shivam");  
System.out.println(s);//manshu
```

**Heap**

(no reference) **manshu shivam**

**SCP**

**s -> manshu**

**shivam**



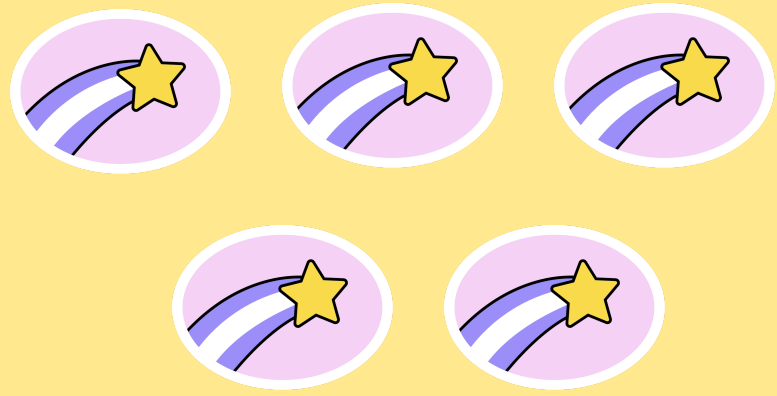


```
String s1 = new String ("manshu")  
String s2 = s1.toUpperCase();  
String s3= s1.toLowerCase();
```

```
System.out.println(s1 == s2);  
System.out.println( s1 ==s3);
```

what will be output ?? -->





most valuable concept



Note -

Because runtime operation

if there is a change in content  
with those changes a new object will be created only  
on the heap but not in SCP.

If there is no change in wrt current content(object).  
no new object will be created  
the same object will be reused.

This rule is same whether object present on the Heap  
or SCP



```
String s1 = new String ("manshu")  
String s2 = s1.toUpperCase();  
String s3= s1.toLowerCase();
```

```
System.out.println(s1 == s2); // false  
System.out.println( s1 ==s3); // true but why ?
```



solution



`String s3 = s1.toLowerCase();` --> wtr s1, s3 content is same





```
String s1 = new String ("manshu")
```

```
String s2 = s1.toUpperCase();
```

```
String s4 = s2.toLowerCase();
```

```
;  
System.out.println( s1==s4); //false why ?
```



solution

(no reference)

**Heap**

**s1 --> manshu**

**s2 --> MANSHU**

**SCP**

**manshu**

String s4 = s2toLowerCase(); --> wtr s2, s4 has different content

**Heap**

**s1 --> manshu**

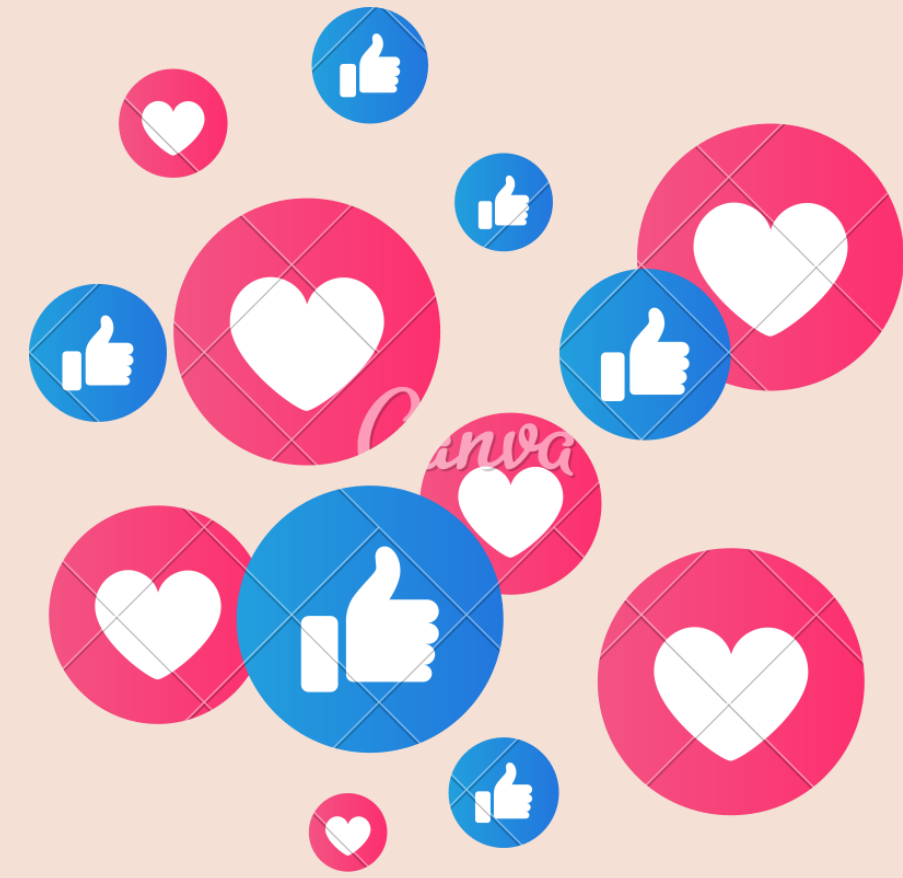
**s2 --> MANSHU**

**s4 --> manshu**

**SCP**

**manshu**

THANK  
YOU



Follow for such content



disclaimer :-  
this slide does not make any  
sence here

in Hindi String is called "dhaga"

and Kabir explains immutability concept like



रहिमन धागा प्रेम का, मत तोड़ो चटकाय।  
टूटे से फिर ना जुड़े, जुड़े गाँठ पर जाय॥

string can not change  
if you change new object will be created