

→ String → collection of characters

→ denoted by `" "` , `"I"`;

Abstr - char
String

→ In java, String is a class

Arrays " " "

'I' → char

I - integer

"I" → String

→ String is Non-primitive Data Type

Classes & Objects

→ Arrays String / Interface
user defined classes

How many ways are there to create String?

→ Way 1: by literal → *jane* is case-sensitive → TEST ①
test ②

→ Way 2 by using new Keyword

a -
A -

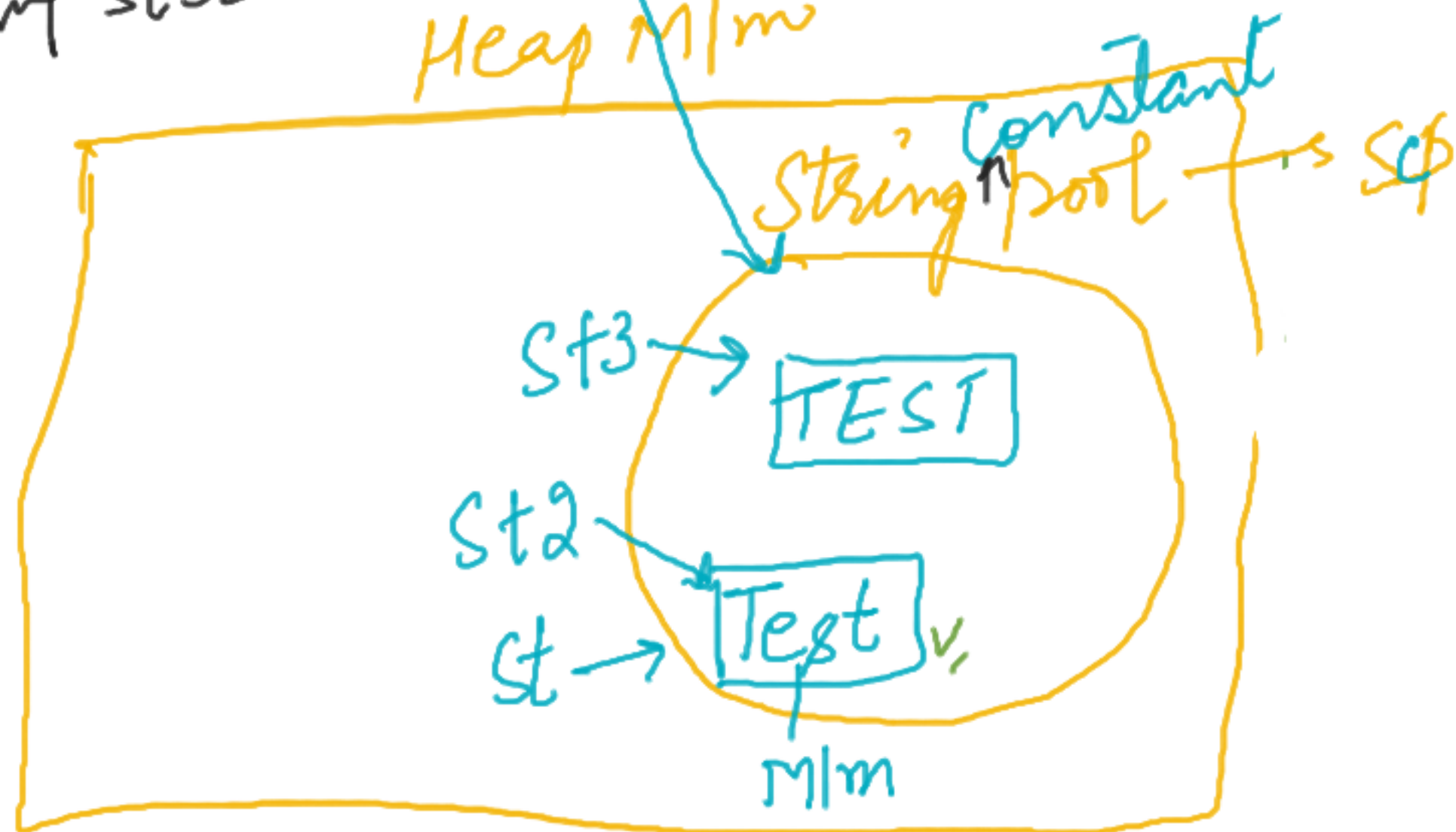
Way 1 → String str = "Jagroop"; ✓✓

Way 2 → String str1 = new String("Test");

Way 1: Literal any var-name
String st = "Test"; str-value

String st2 = "Test"; → SCP
objects

String st3 = "TEST"
Heap M/m



How many object will be

create if we create the
string by using literal?

→ ① → M/m → SCP

Way 2 :- Creation of String using 'new' keyword → Object

String st = new String ("Test String");

↓ ↓ ↓ ↓ ↓
<Class> any var Key- Class ↓
name word Name value.

How many objects will be created?

→ ② M/m allocation will be in Heap M/m =

create one unreference object in SCP also.

② String s1 = new String("Test");

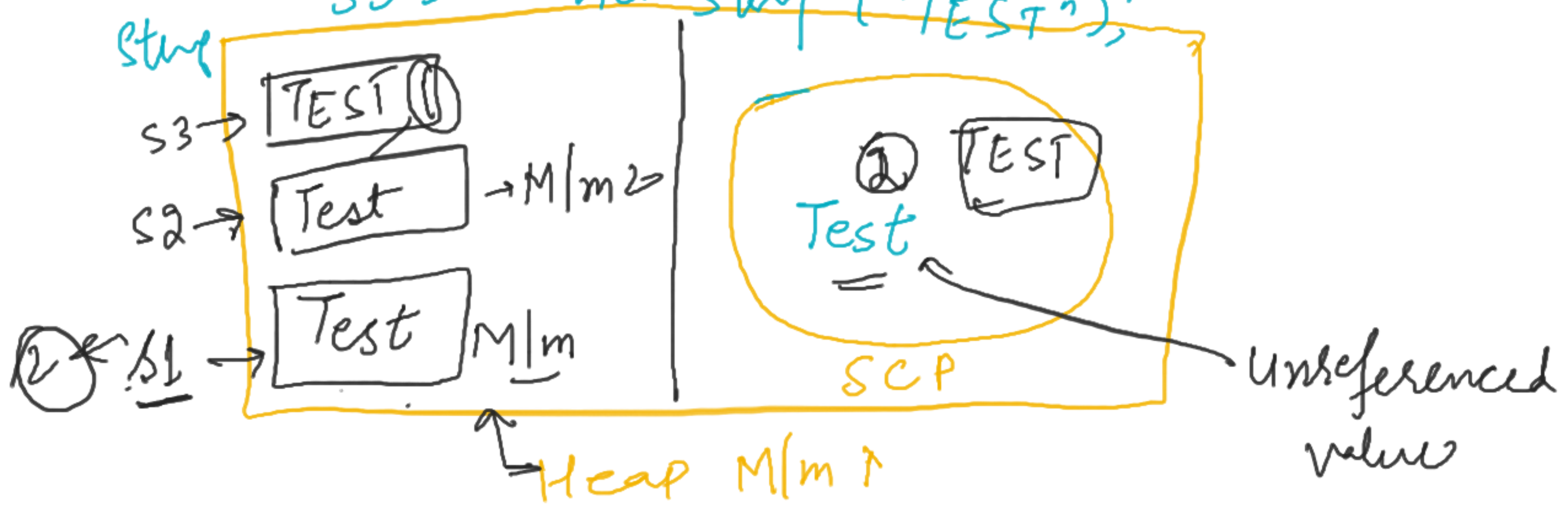
Object 1

Object

creation of object by using new keyword

String s2 = new String("Test");
String s3 = new String("TEST");

→ Object 2



Student

Teacher \xrightarrow{n} Student
 \checkmark $M/madd \rightarrow Id$
Teacher 2

SCP
Admin
Ref.
Student

Inst/college

Why String is immutable?

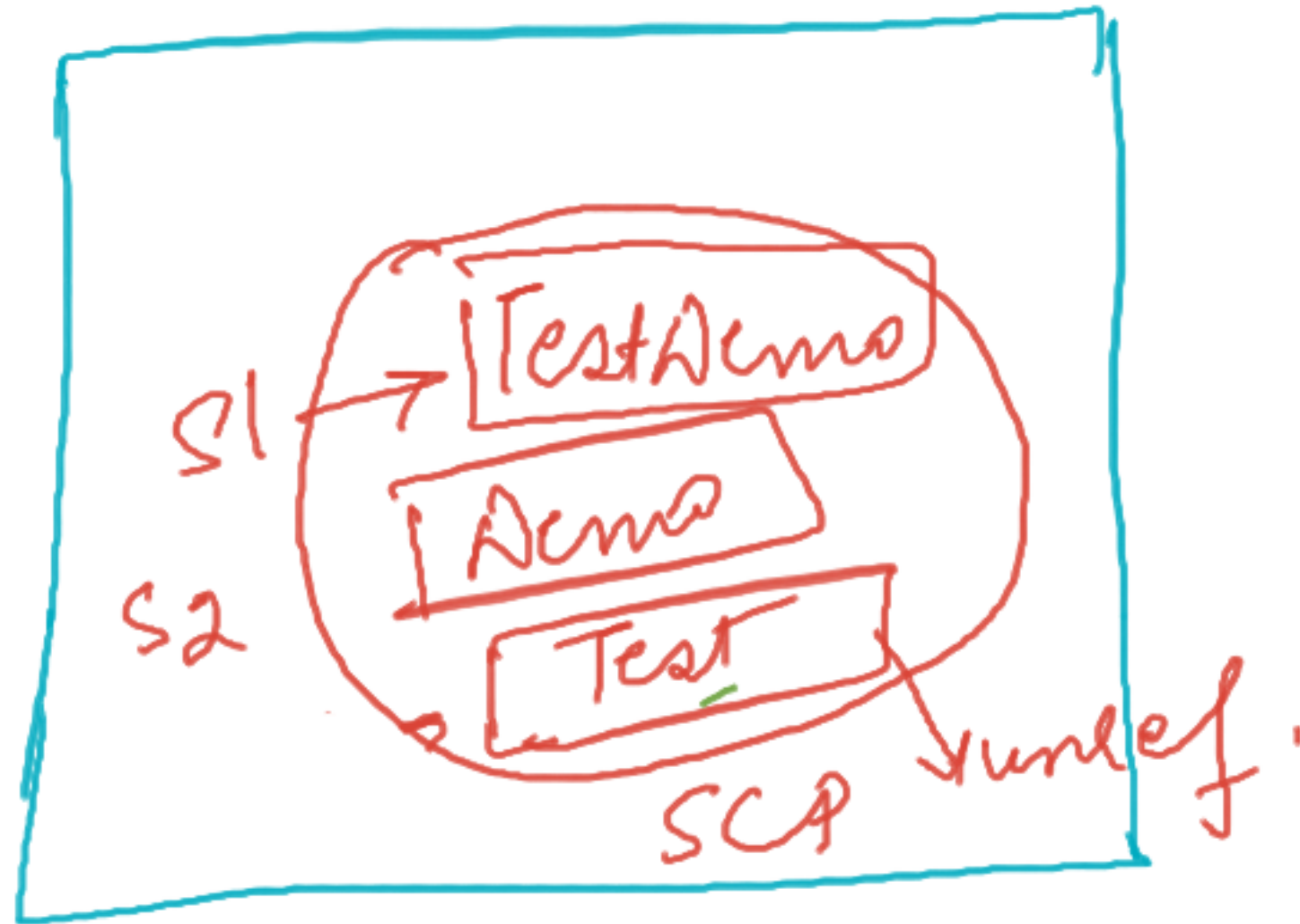
Once we create an object of string, we can't change its value.

```
String S1 = "Test";
```

```
S1 = "Test123";
```


int i = 10;
i = 100;

i = ~~10~~ 100



Heap M/m

String s1 = "Test";

String s2 = "Demo";

sys0 (s1.concat(s2););

// Test Demo →

s1 = s1.concat(s2);

arr.length

st.length();

st = Testing

↓
0 1 2 3 4 5 6
↓
T e s t i n g

Index No.

st.charAt(0)

st.charAt(1)

T
e
s
t
i
n
g

Start
←
Testing → 7
↓
0
↓
4
↓
6

→
Array
= 0

T	e	s	t	i	n	g
0	1	2	3	4	5	6