**java.lang package – Part-14**

* **More about String Heap & SCP:**

class Test{

public static void main(String[] args){

String s1 = new String(“you cannot change me!”);

String s2 = new String(“you cannot change me!”);

System.out.println(s1 == s2); //false

String s3 = “you cannot change me”;

System.out.println(s1 == s3); // false

String s4 = “you cannot change me”;

System.out.println(s3 == s4); //true

String s5 = “you cannot”+”change me!”; // (1)

System.out.println(s3 == s5);

String s6 = “you cannot”;

String s7 = s6+”change me!”; // (2)

System.out.println(s3 == s7); // false

final String s8 = “you cannot”;

String s9 = s8 + “change me!”;

System.out.println(s3==s9); // true

System.out.println(s6==s8); // true

}

}

Line\_1:

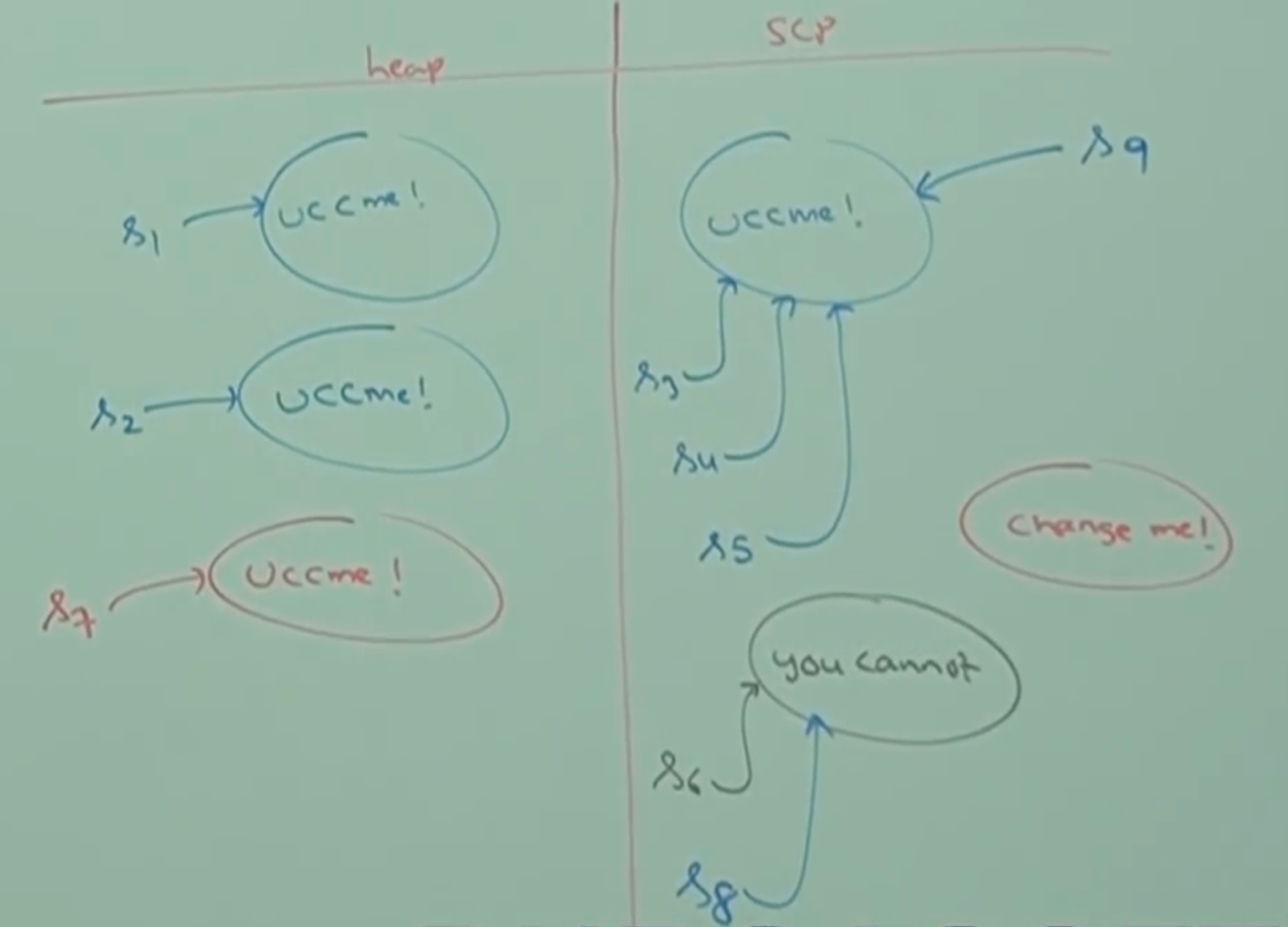
This operation will be performed at compile-time only because both arguments are compile-time constants.

Line\_2:

This operation will be performed at runtime only because at least one argument is normal variable.

Line\_3:

This operation will be performed at compile-time only because both arguments are compile-time constants



* **Interning of String Objects:**

We can use intern() method to get corresponding SCP object reference by using head object reference.

By using heap object reference if we want to get corresponding SCP object reference. Then, we should go for intern() method.

Example:

class Test{

public static void main(String[] args){

String s1 = new String(“durga”);

String s2 = s1.intern();

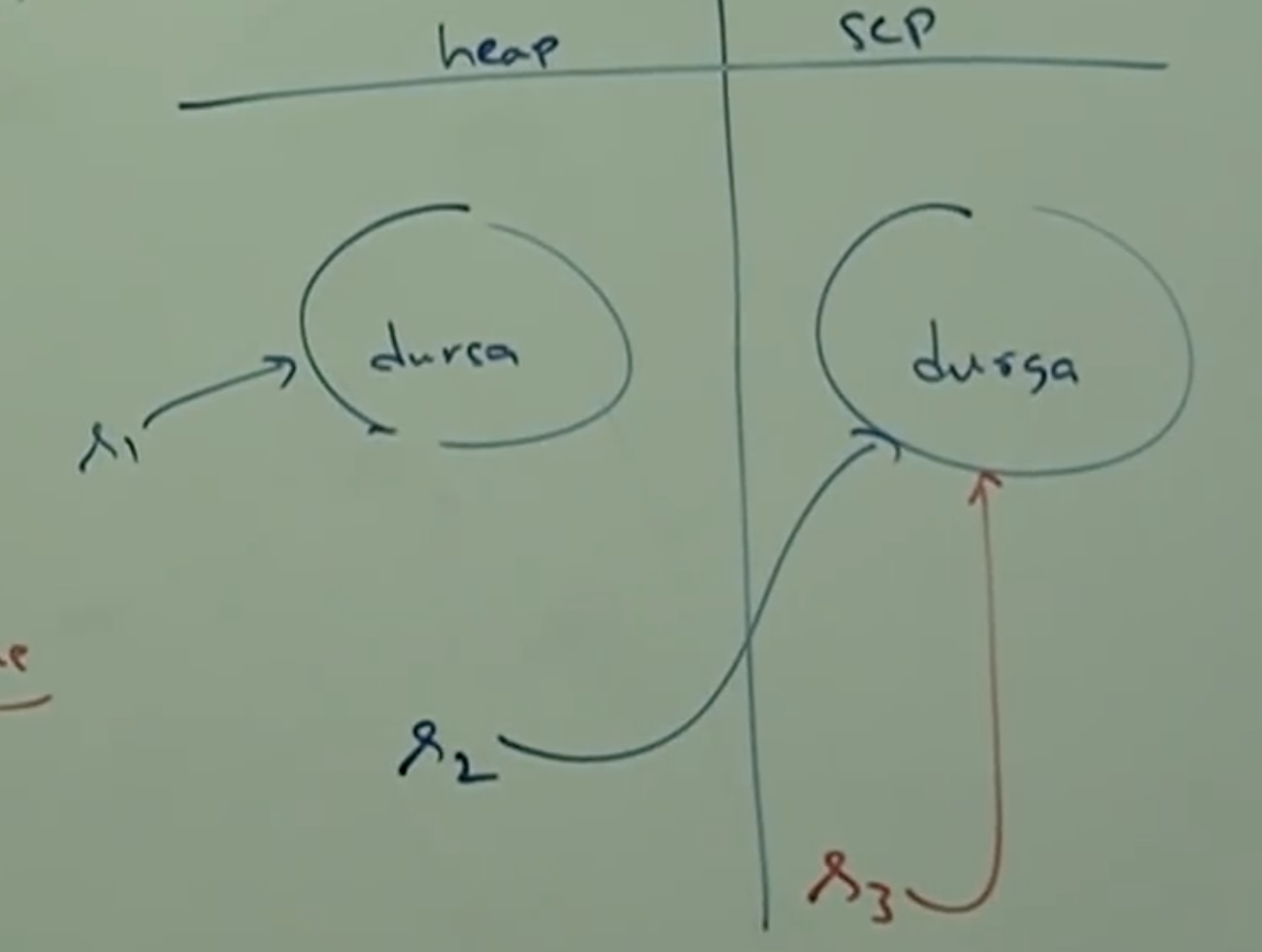
System.out.println(s1 == s2); //false

String s3 = “durga”;

System.out.println(s2==s3); //true

}

}



If the corresponding SCP object is not available then intern method itself will create the corresponding SCP object.

Example:

class Test{

public static void main(String[] args){

String s1 = new String(“durga”);

String s2 = s1.concat(“software”);

String s3 = s2.intern();

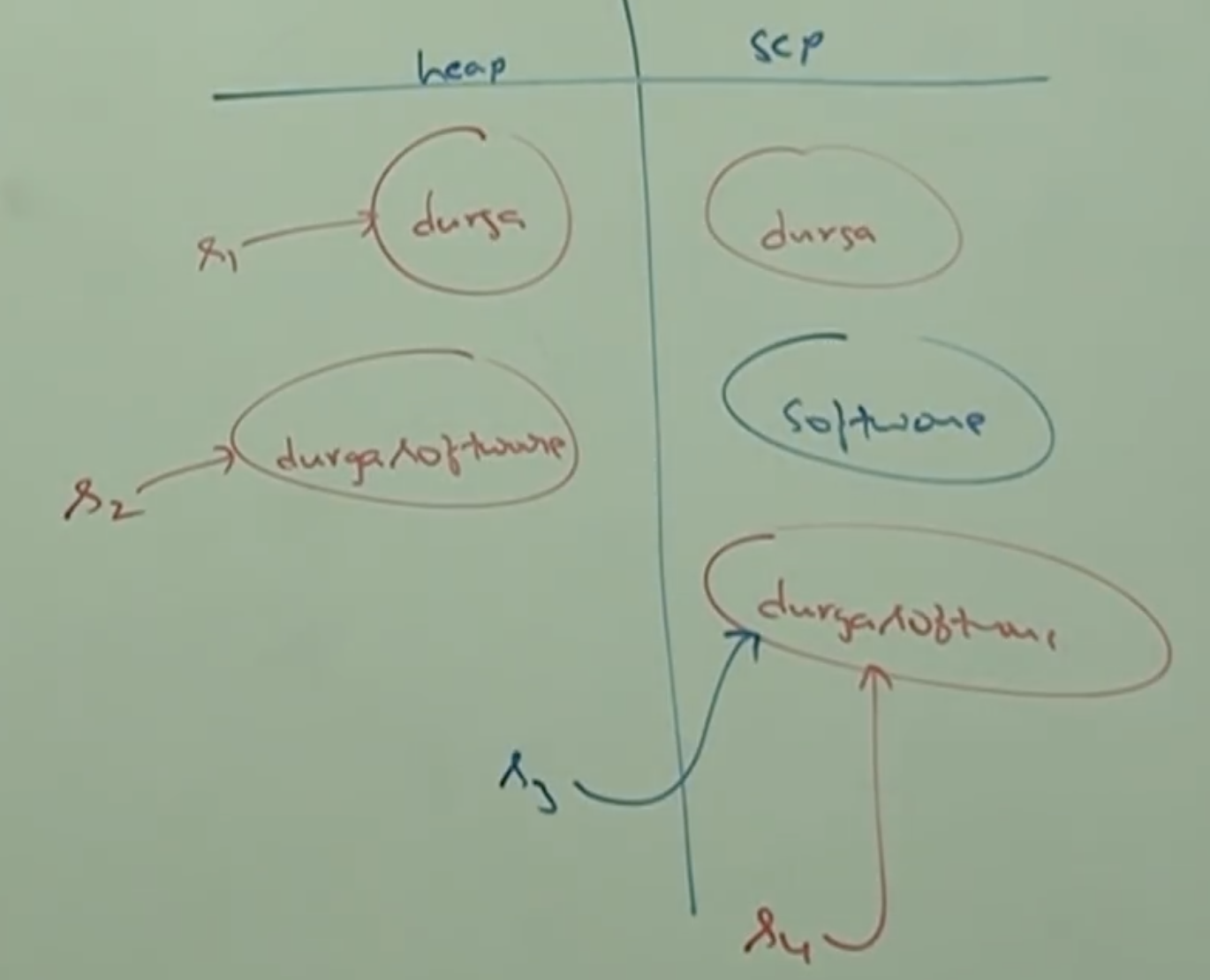
System.out.println(s2 == s3); // false

String s4 = “durgasoftware”;

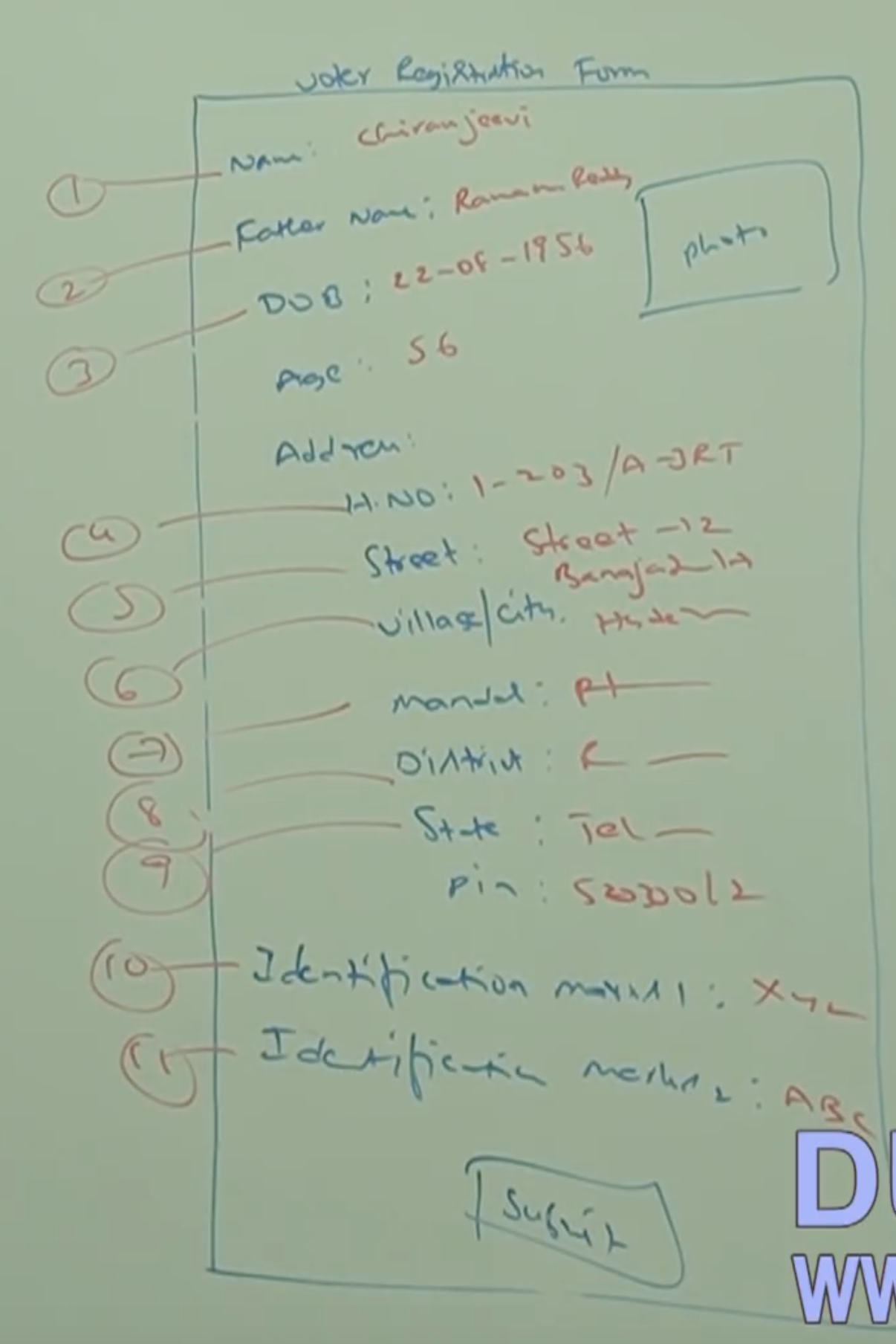
System.out.println(s3 == s4); // true

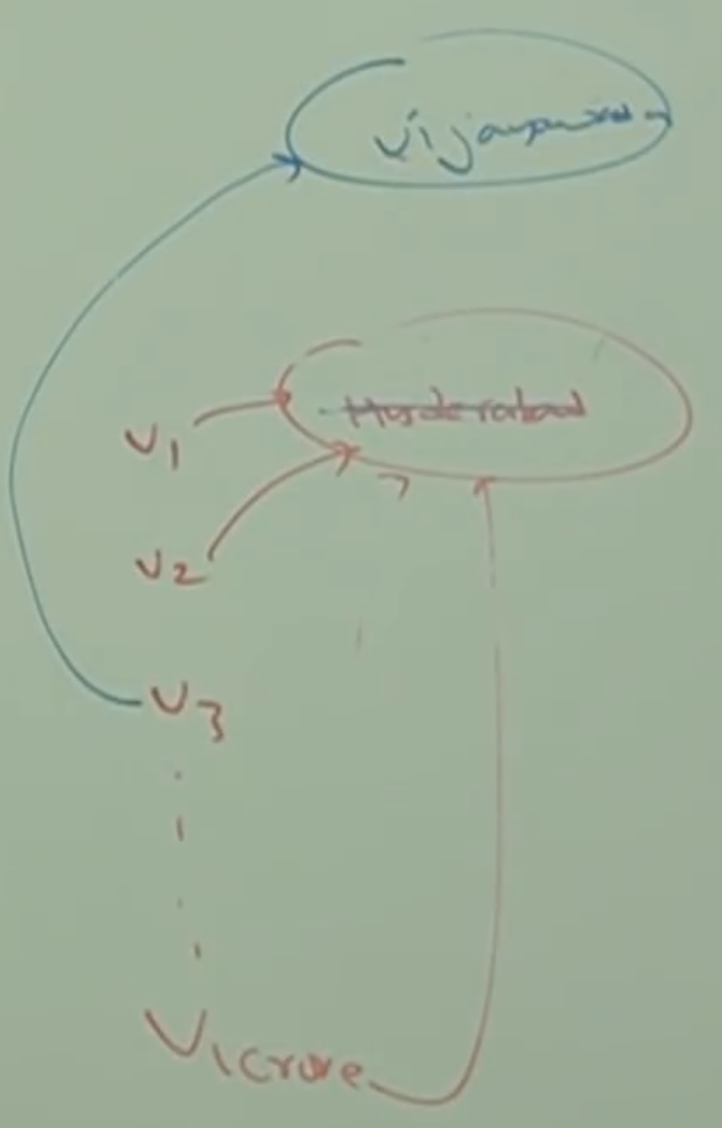
}

}



* **Importance of String Constant Pool:**





In our program if a String object is repeatedly required then it is not recommended to create separate object for every requirement. Because, it creates performance and memory problems.

Instead of creating a separate object for every requirement we have to create only one object and we can re-use the same object for every requirement, so that performance and memory utilization will be improved.

This thing is possible because of SCP. Hence, the main advantages of SCP are memory utilization and performance will be improved.

But the main problem with SCP is, as several references pointing to the same object, by using one reference if we are trying to change the content then remaining references will be affected.

To overcome this problem SUN people implemented String objects as immutable. That is once we create a String object we can’t perform any changes in the existing object.

If we are trying to perform any changes with those changes a new object will be created. Hence, SCP is the only reason for immutability of String objects.

* **FAQ’s:**

1. What is the difference between String and StringBuffer?
2. Explain immutability and mutability with an example.
3. What is the difference between

String s = new String(“durga”);

String s = “durga”;

1. Other than immutability and mutability is any other difference between String and StringBuffer?

In String .equals() method meant for

1. What is SCP?

It is a specially designed memory area for String objects.

1. What is the advantage of SCP?

Memory utilization and performance.

1. What is the disadvantage of SCP?

Immuatability

1. Why SCP like concept is available only for String but not for StringBuffer?

String is the most commonly used object and hence SUN people provided special memory management for String objects.

But StringBuffer is not commonly used object and hence special memory management is not required for StringBuffer.

1. Why String objects are immutable whereas StringBuffer objects are mutable?

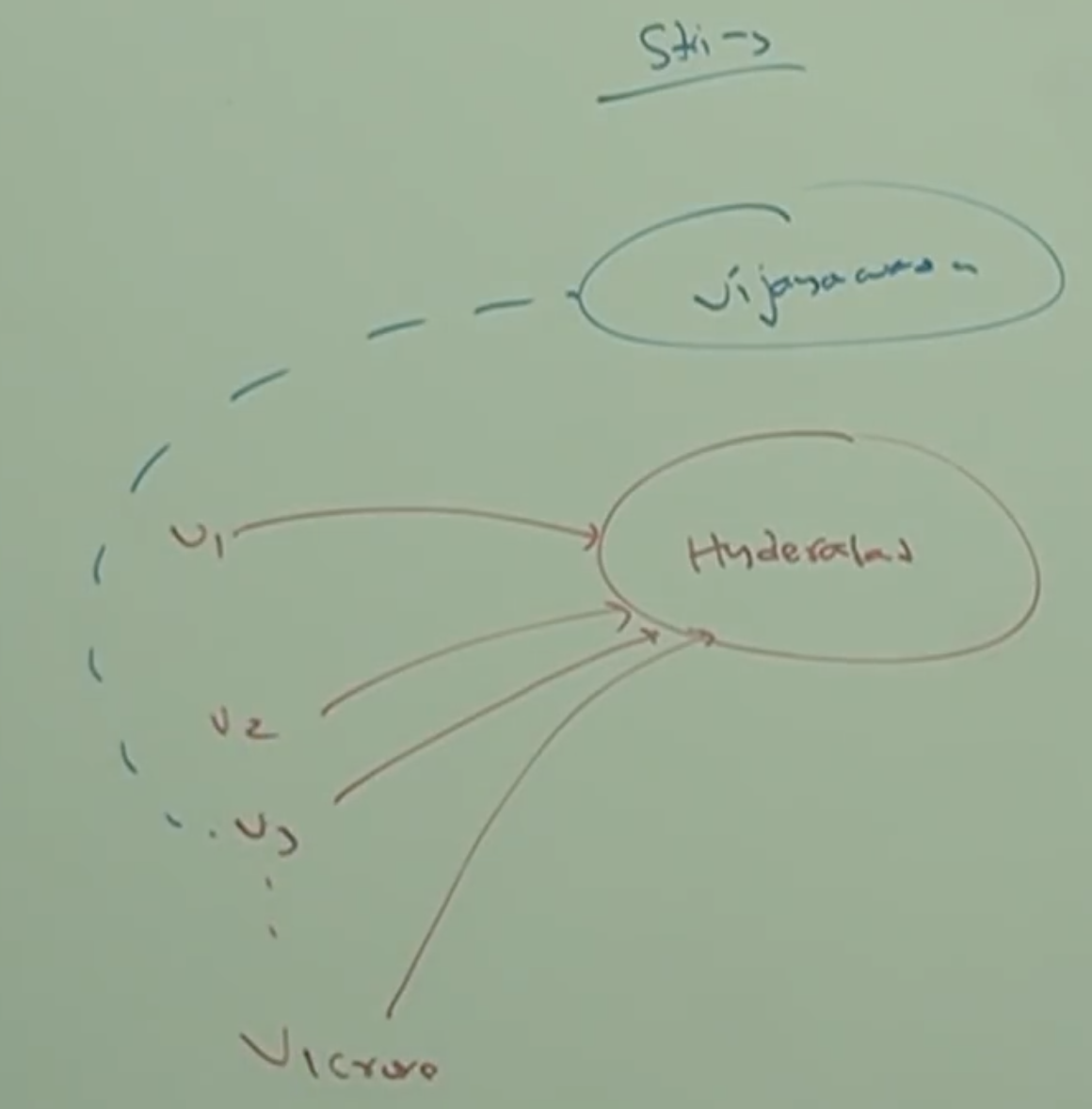
In the case of String, because of SCP a single object can be referenced by multiple references.

By using one reference if we are allowed to change the content in the existing object, then remaining references will be affected. To overcome this problem, SUN people implemented String objects as Immutable.

According to this, once we create a String object we can’t perform any changes in the existing object.

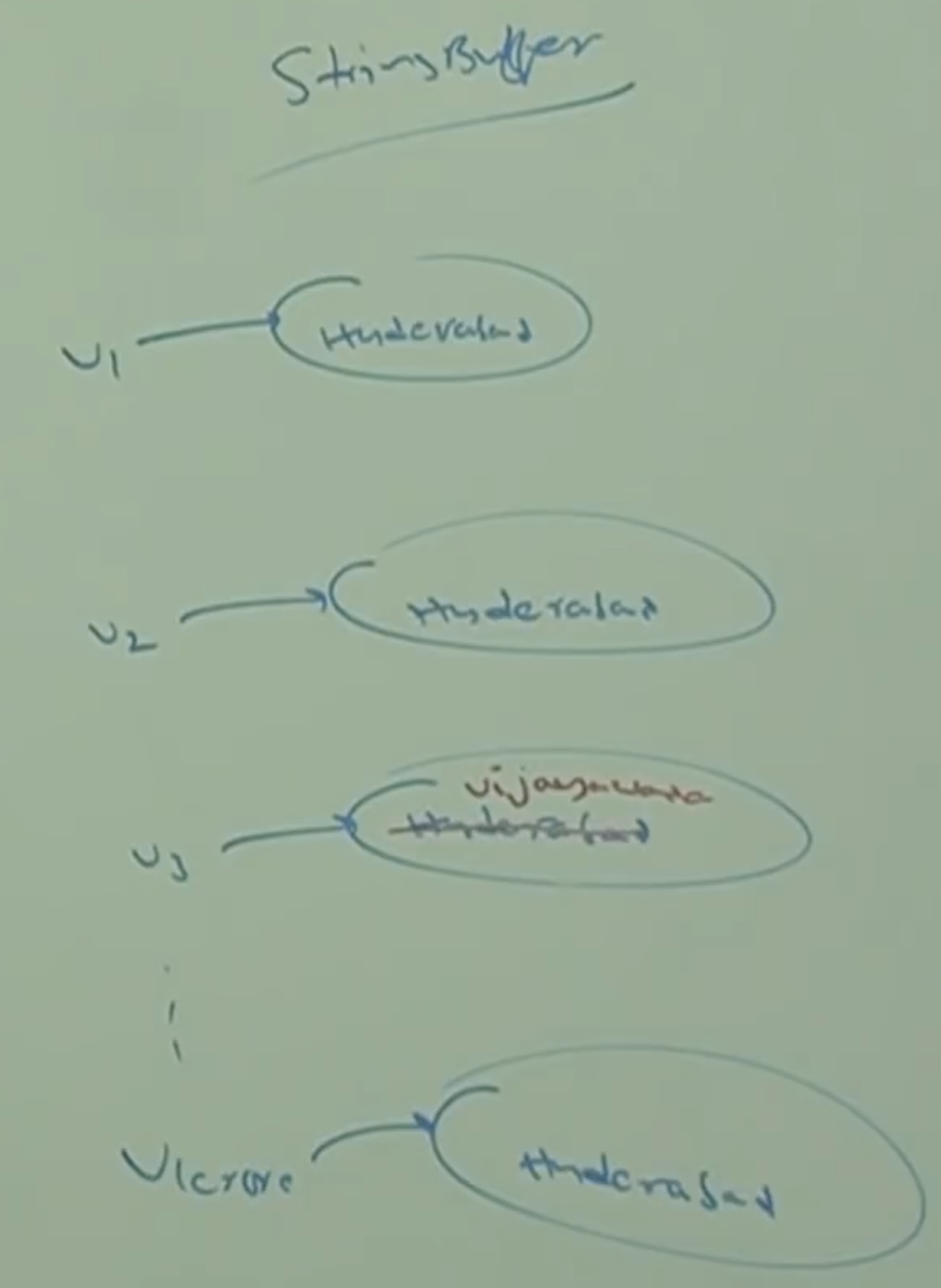
If we are trying to perform any changes with those changes a new object will be created.

String:



StringBuffer:

But in StringBuffer there is no concept like SCP hence for every requirement a separate object will be created. By using one reference if we are trying to change the content then there is no effect on remaining references. Hence, immutability concept not required for the StringBuffer.



1. In addition to String objects any other objects are immutable in Java?

Yes, all the Wrapper class objects also immutable.

1. Is it possible to create our own immutable class?

Yes

1. How to create our own immutable class? Explain with an example.
2. Immutable means non-changeable whereas final means also non-changeable. Then what is the differene between final and immutable?