String class

- A **java.lang.String class** is final which implies no class can extend it

String is Immutable class

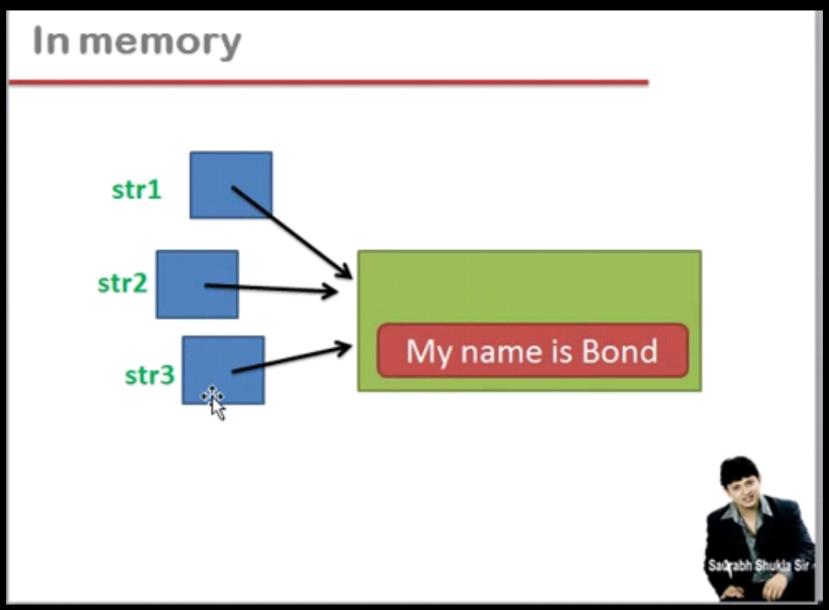
- **Java String Class** is immutable, i.e. **Strings in java,** once created and initialized, cannot be changed on the same reference.

Creating String Object

- A simple String can be created using a string literal enclosed inside double quotes as shown
- String str1 = “My name is Bond”;

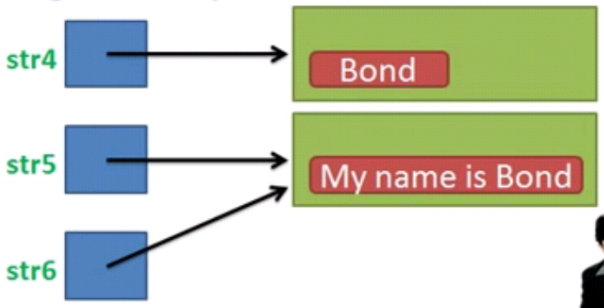
Important Point

- If two or more Strings have the same set of characters in the same sequence then they share the same reference in memory
- String str1 = "My name is Bond";
  String str2 = "My name is Bond";
  String str3 = "My name "+ "is Bond";
- All the String references str1, str2 and str3 denote the same String object


In memory
str1
str2
str3
My name is Bond
Saurabh Shukla Sir

...In Memory

- String str4 = "Bond";
  String str5 = "My name is "+ str4;
  String str6 = "My name is Bond";



Creating String with new keyword

- String str5 = new String("My name is Bond");

Concatenation Operator

- The java.lang.String class differs from other classes, one difference being that the String objects can be used with the += and + operators for concatenation

Output

```
class StringExample1{
 public static void main(String[] args){
  String s1="computer";
  String s2="computer";
  String s3=new String("computer");
  System.out.println("Result 1:"+(s1==s2)); //true
  System.out.println("Result 2:"+s1.equals(s2)); //true
  System.out.println("Result 3:"+(s1==s3)); //false
  System.out.println("Result 4:"+s1.equals(s3)); //true
 }
}
```