Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

String Introduction in Java

=> String :-

➔ String is the sequence of characters or say String is an array of characters. For example : char[] c={'d', 'e', 'e', 'p', 'a', 'k'};
   o To handle the character sequence, java has provided an interface i.e. CharSequence

➔ String is a non-primitive data type.

➔ To create String or to perform String operations, java has provided some predefined classes :-

1. java.lang.String

2. java.lang.StringBuffer

3. java.lang.StringBuilder

4. java.util.StringTokenizer

➔ Syntax :

public final class String extends Object
implements Serializable, Comparable,
CharSequence
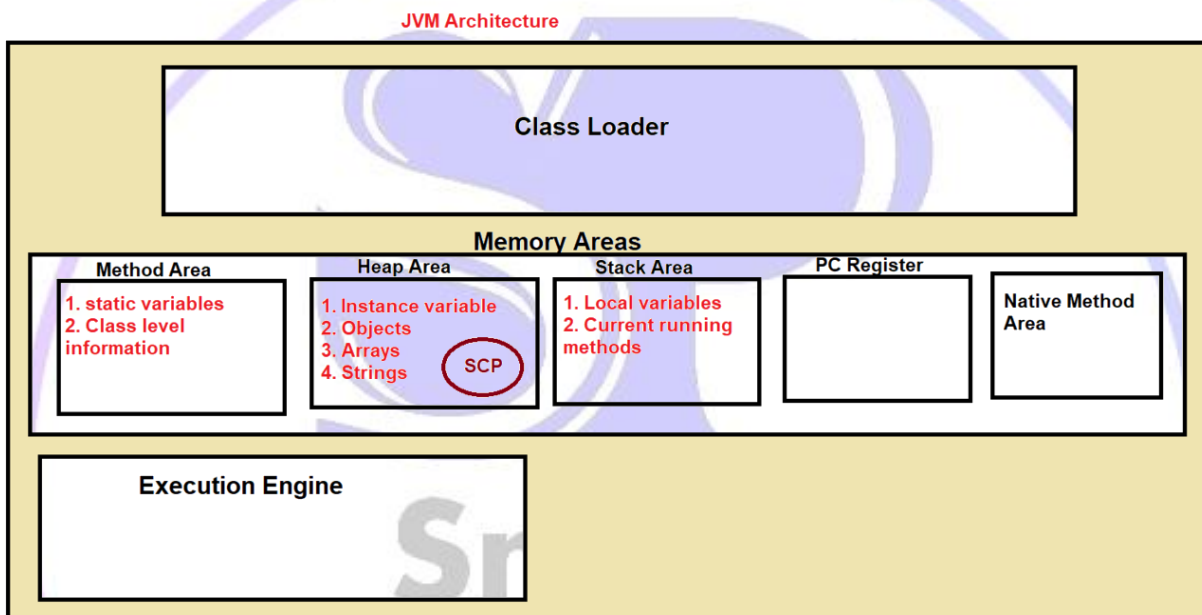
    {

        //methods

    }

➔ String is the class and we can create String
class object. But we can create String class
object by 2 ways :-
1. String str=new String("deepak");
2. String str="deepak";

➔ Whenever we create String class object, objects
created are "IMMUTABLE"

➔ Whenever we create String objects, it allocates memory in special memory area i.e. "String Constant Pool" or "String Literal Pool"

JVM Architecture

Class Loader

Memory Areas

Method Area	Heap Area	Stack Area	PC Register	Native Method Area
1. static variables 2. Class level information	1. Instance variable 2. Objects 3. Arrays 4. Strings (SCP)	1. Local variables 2. Current running methods		

Execution Engine

➔ Garbage collection is not appolicable for String Constant Pool

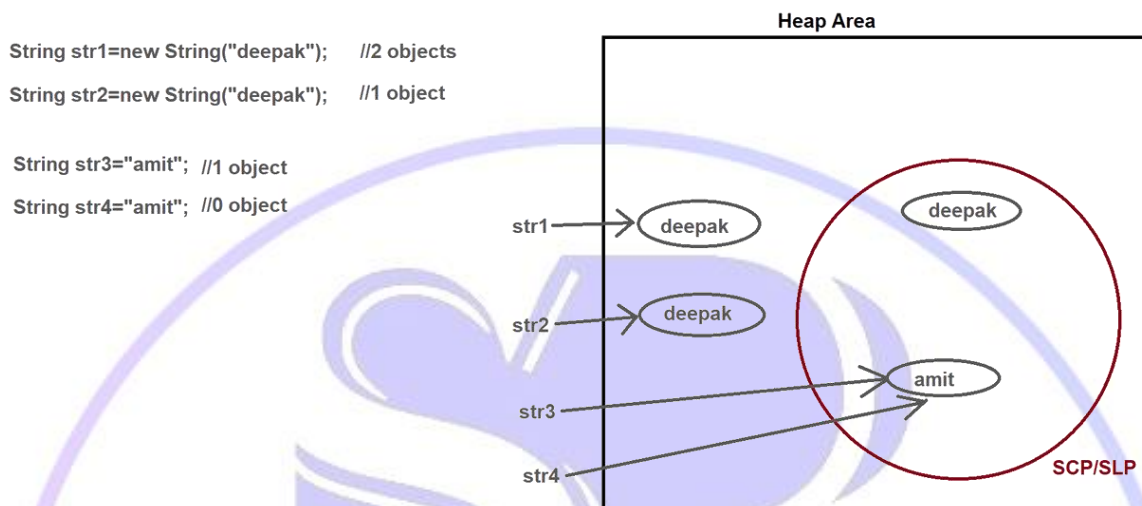## => Difference between creating String objects by "new keyword" and by "String literal"

➔ If we create String object by using new keyword then an object is created in heap area. If we have provided any string literal in string constructor then 2 objects will be created and second object will be created in String Constant Pool

➔ If we create String object by using String literal then an object is created in String Constant Pool

```
1)  String name=new String("deepak");   // 2 Objects

2)  String name="amit";     // 1 Object
```

Heap Area

name → (deepak)

(deepak)

(amit)

name

JVM will create reference variable and that will point to this object

SCP/SLP

```
String str1=new String("deepak");    //2 objects
String str2=new String("deepak");    //1 object

String str3="amit";  //1 object
String str4="amit";  //0 object
```

Heap Area

str1 ───→ deepak

str2 ───→ deepak

deepak

amit

str3

str4

SCP/SLP
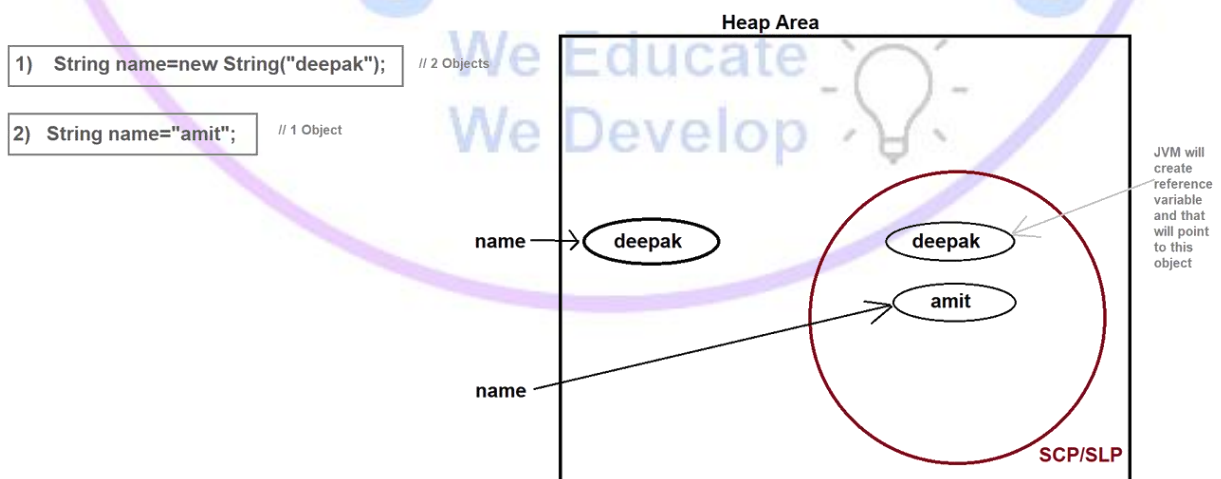
=> Properties of "String Constant Pool" or "String Literal Pool"

1. SCP stores the String Literal Objects

2. Whenever String Literal Object is created in SCP, first it will check wether that literal object is already present in SCP or not, if it is not present then it will create new object otherwise it will not create new object and that reference variable will point to that same object

3. Garbage Collection is not applicable in SCP

=> Why String objects are immutable ?

➔ String objects are immutable which means that if we create any String object, then we cannot change its value

➔ String objects are immutable because if we change any string object value, then it will create new object and will not affect other reference variables

➔ NOTE : Strings are not immutable, String objects are immutable

```
public class Test1
{
    public static void main(String[] args)
    {
        String name="deepak";

        //name=name.concat(" java");

        name.concat(" java");

        System.out.println(name);
    }
}
```
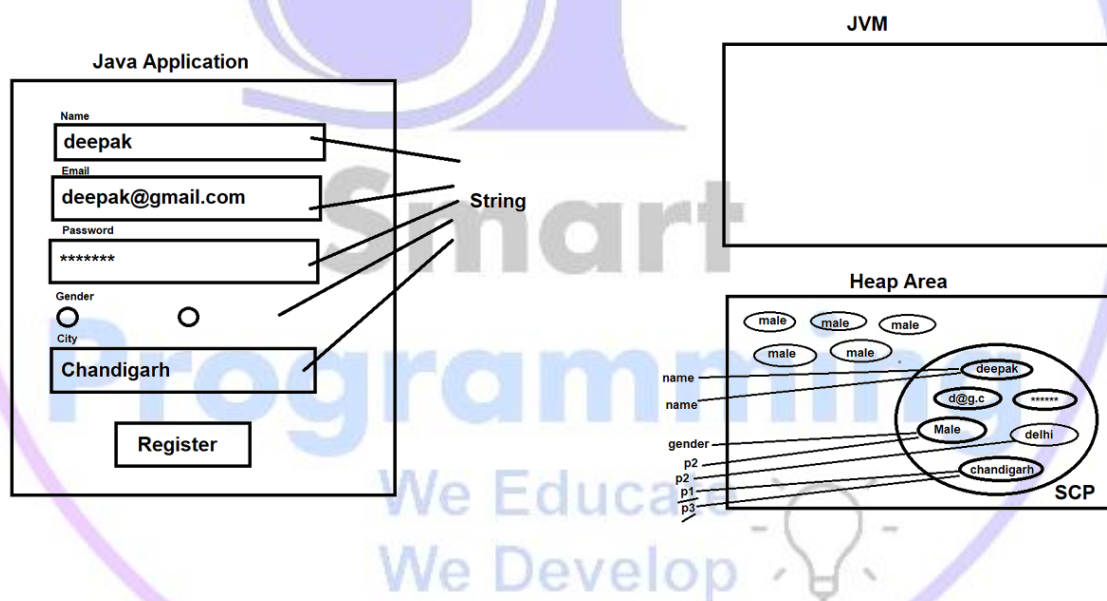
Heap Area

name ──────✗──────→ deepak

java

deepak java

SCP / SLP
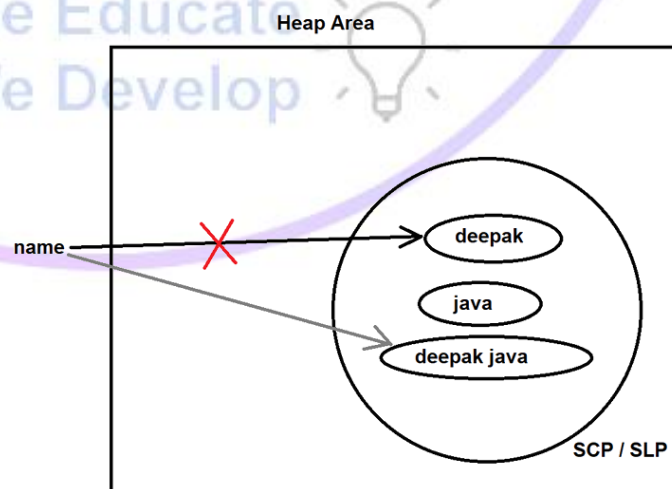
=> What String class is "final" ?

-> String class is final because we cannot inherit String class and thus we were not able to change the String class properties

Real world use of String final class and string immutable object

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming