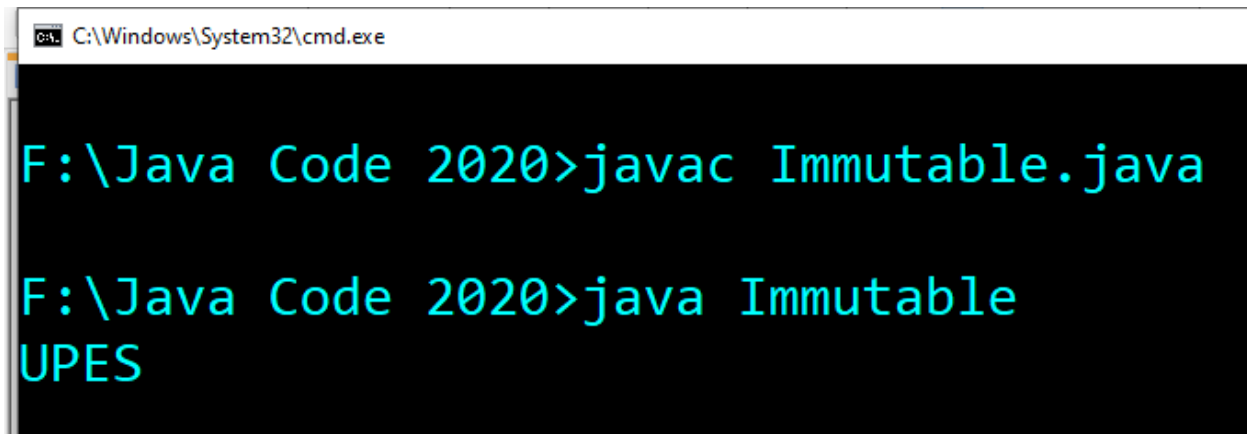# Case Study based on String Objects, SCP and Heap Area

Program Code:

```
1.  class Immutable
2.  {
3.  public static void main(String args[])
4.  {
5.  String s=new String("UPES");
6.  s.concat("College");
7.  System.out.println(s);
8.  }
9.  }
```

```
C:\Windows\System32\cmd.exe

F:\Java Code 2020>javac Immutable.java

F:\Java Code 2020>java Immutable
UPES
```

Note:
Once we create a string object, we cannot make any changes to the existing object. If we are trying to make a change, a new object will be created with those changes. This behavior is called **immutability** of the string object.

Note: For every new operator one Sting object will be created in heap and for every string literal one copy will be created in SCP for reference.
Note: Because of any run time operation or method call if a new string object is required to create, compulsorily this object will be created in heap area only not in SCP area.
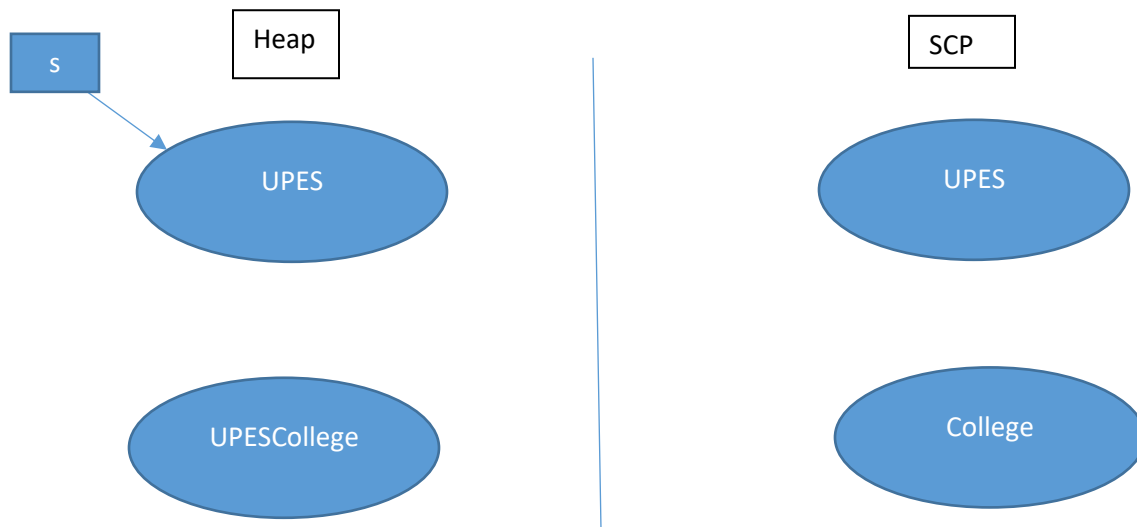
Question:
In the above code, how many objects will be created in Heap and SCP area and what are their names?

Answer:
Line no 5: An object named UPES(ref s) will be created in the heap and an object named UPES (copy) will be created in SCP for future purpose.
Line no 6: College is a string literal, so that an object named college will be created in SCP and an object named UPESCollege will be created in the heap area due to a run time operation or method call (concat).

Question:
Do we have any reference variables for UPESCollege that present in heap memory?
Answer:
 No, so it will be eligible for Garbage Collection.
Question:
 Do we have any reference variables for UPES and College objects present in SCP area?
Answer:
Yes, JVM internally maintains references to SCP objects. Therefore they are not eligible for garbage collection.
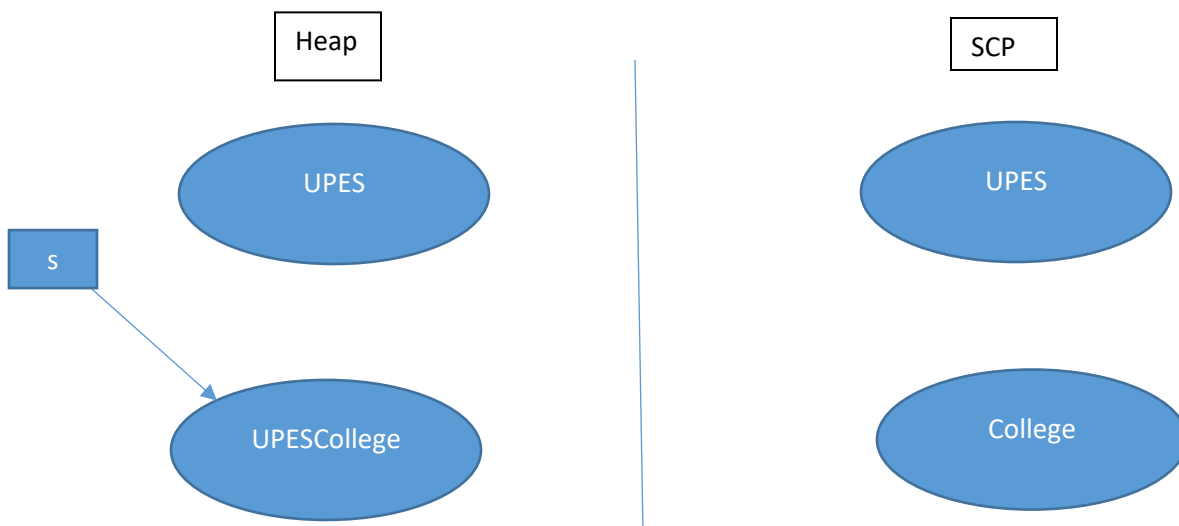
Program Code:

```
1.  class Immutable
2.  {
3.  public static void main(String args[])
4.  {
5.  String s=new String("UPES");
6.  s=s.concat("College");
7.  System.out.println(s);
8.  }
9.  }
```

```
F:\Java Code 2020>java Immutable
UPESCollege
```

Question:
In the above code, how many objects will be created in Heap and SCP area and what are their names?
Answer:

Heap	SCP
UPES	UPES
s → UPESCollege	College

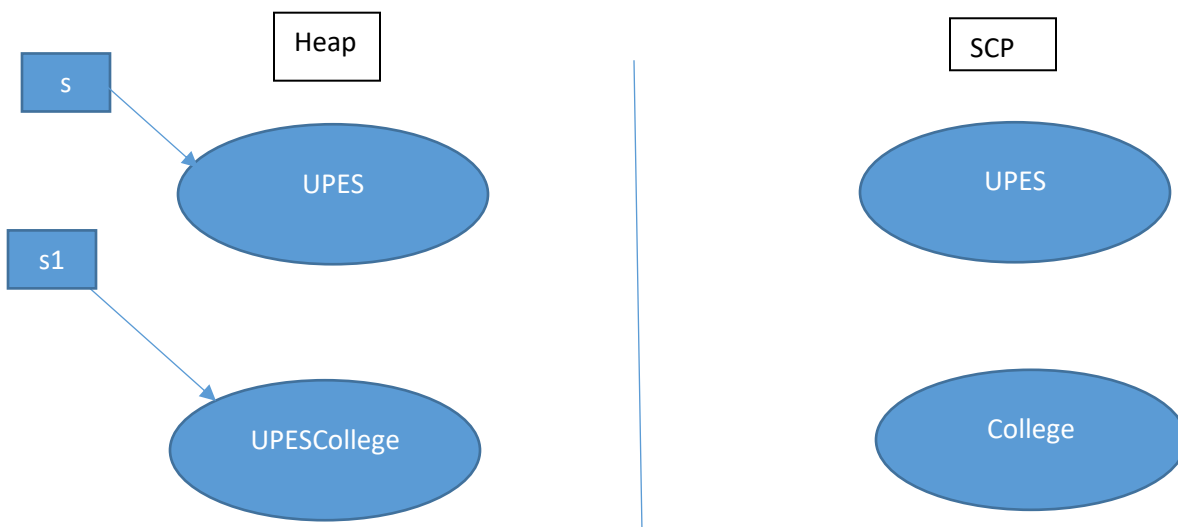Object Oriented Programming (CSEG2016)

Program code:

```
public class Immutable
{
public static void main(String args[])
{
String s=new String ("UPES");
String s1=s.concat("College");
System.out.println(s);
System.out.println(s1);
}
}
```

```
F:\Java Code 2020>java Immutable
UPES
UPESCollege
```

Question:
In the above code, how many objects will be created in Heap and SCP area and what are their names?
Answer:

Program Code:

```
01.  class Test
02.  {
03.      public static void main(String[] args)
04.      {
05.          String s1 = new String("Spring");
06.          s1.concat("Fall");
07.          String s2 = s1.concat("Winter");
08.          s2.concat("Summer");
09.          System.out.println(s1);
10.          System.out.println(s2);
11.      }
12.  }
```

Questions:
What is the output we are going to get?
How many total object will be created?
How many object will be created in Heap and SCP and what will be their names?


Answers:

sop(s1) –Spring
sop(s2)-SpringWinter
Total object: 8(4 in Heap+4 in SCP)

Line 5 ; 2 object: Spring(s1 ref var) in heap and copy of spring in scp for future purpose.
Line 6: Because of run time operation or method call SpringFall in heap and Fall in scp
Line 7 Springwinter (s2 ref var) in heap and winter in scp
Line 8:Summer in scp due to string literal and SpringWinterSummer in heap due to run time operation
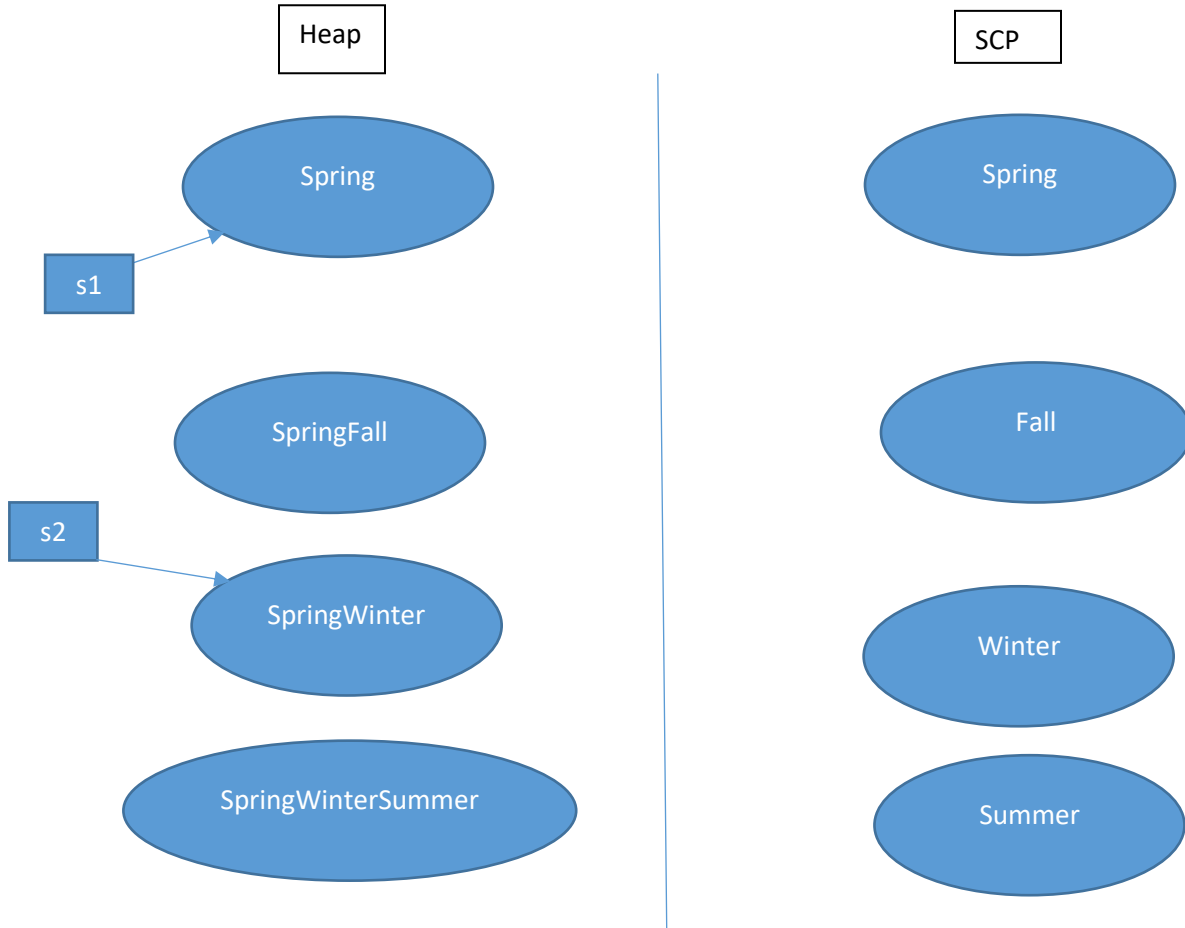
Note: For Springfall and SpringWinterSummer ,we don't have any reference so they are eligible for GC but object presents in SCP are not eligible for GC because internally JVM is maintained the references for SCP Objects.

Program Code:
1. String s1= "Spring";
2. String s2= s1+ "Summer";
3. s1.concat("Falls");
4. s2.concat(s1);
5. s1+ = "Winter";
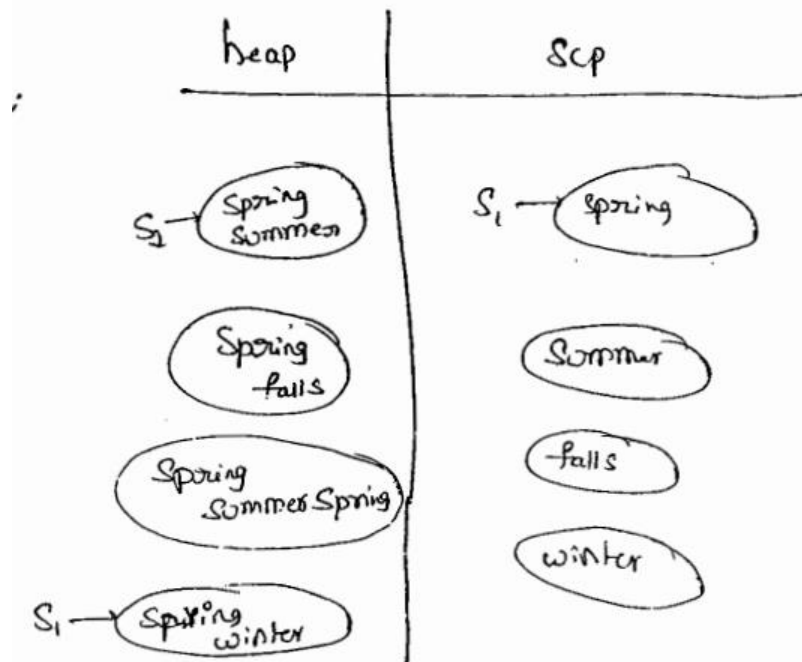6. SOP(s1);//springwinter
7. SOP(s2);//springsummer

Question:
In the above code, how many objects will be created in Heap and SCP area and what are their names?
Answer:

Total 8 object will be created, 4 in Heap area and 4 in SCP area.
1. An object named Spring will be created in SCP area that will be pointed by s1.
2. one object named Springsummer(ref s2)j will be created in Heap area and 1 object named Summer will be created in SCP.
3. falls in SCP and Springfalls in heap.
4. SpringSummerSpring in heap
5. winter in SCP and Springwinter(now ref by s1) in Heap.

Output:
SpringWinter
SpringSummer

Program Code:

String s = new String("UPES");
s.concat("Computer");
s=s.concat("Lab");

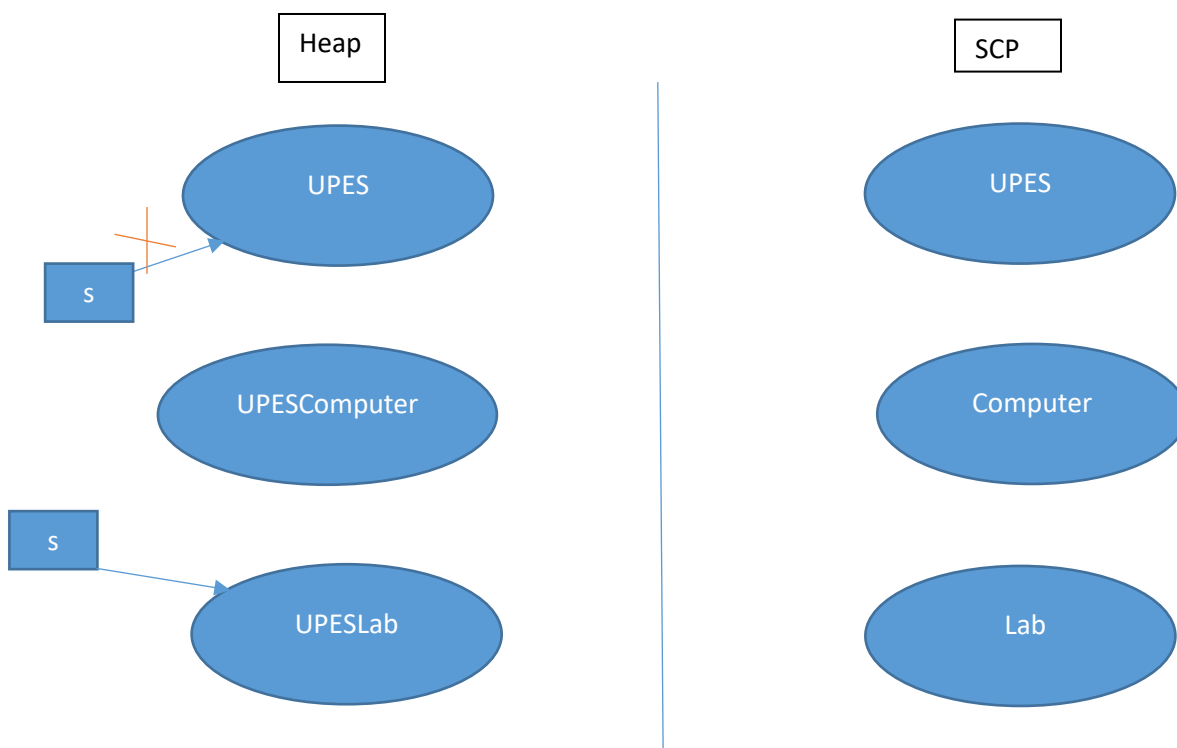Total, how many object will be created? how many in Heap and how many in SCP?

Answer:

1. One object named UPES(ref s) will be created in heap and copy of UPES in SCP for future purpose.
2. "Computer " in SCP because it is a string literal and for every string literal one copy will be created in SCP.Due to run time operation or method call (concat) one object named UPESComputer will be created in Heap area.
3. One object named Lab will be created in SCP and object UPESLab will be created in Heap because of method call and now onwards s will refer to it.

Question: Do we have any reference variables for UPES and UPESComputer present in heap memory?

Answer: No, so they will be eligible for Garbage Collection.

Question: Do we have any reference variables for UPES, Computer, and Lab objects present in SCP area?

Answer: Yes, implicit reference variables are maintained by the JVM so that they are not eligible for garbage collection.
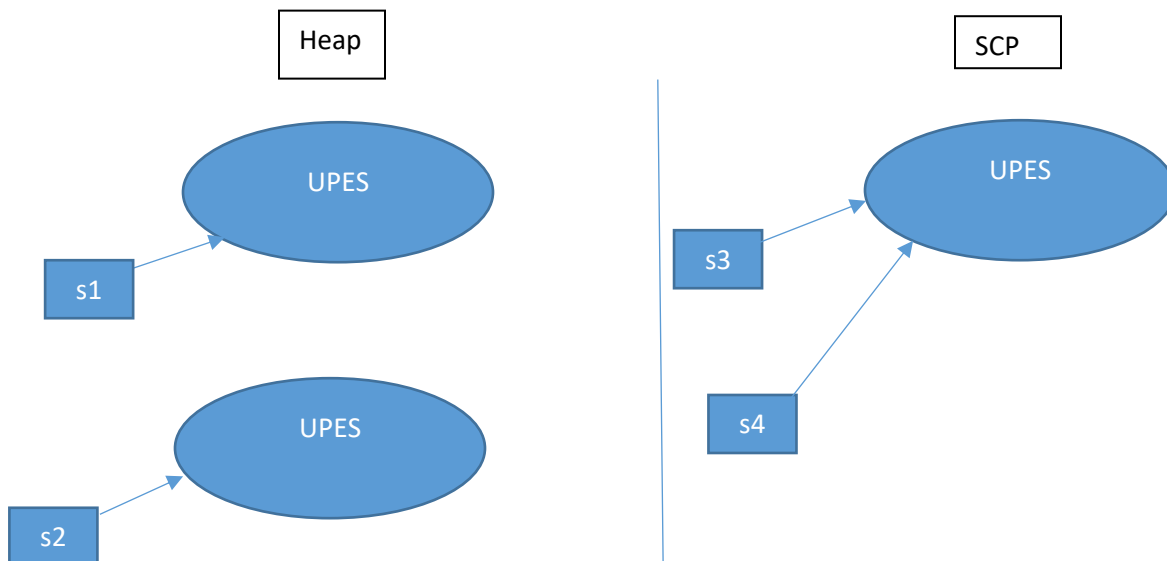
Program Code:

String s1=new String ("UPES");
String s2=new String ("UPES");
String s3= "UPES";
String s4= "UPES";

Question:
In the above code, how many objects will be created in Heap and SCP area and what are their names?
Answer:

Heap

UPES

s1

UPES

s2

SCP

UPES

s3

s4

Note: There may be chance of existing 2 objects with same content in heap area, but there is no chance of existing 2 objects with same content in SCP area,

Object Oriented Programming (CSEG2016)