SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Object Oriented Programming using Java Laboratory (DJS22FEL22)

Name – Sujal Vivek Choudhari   **Sap id** - 60003220216   **Roll No.** – I167

6. To implement Constructors and constructor overloading

a. WAOOP to count the no. of objects created of a class using constructors

Code:

```java
J Expriment6A.java > 🍃 Expriment6A > ⊕ main(String[])
1    class TestClass {
2        public static int count;
3        TestClass(){
4            count++;
5            System.out.println(x: "New Class Created");
6        }
7    }
8
9
10   public class Expriment6A {
         Run | Debug
11       public static void main(String[] args) {
12
13           new TestClass();
14           new TestClass();
15           new TestClass();
16           new TestClass();
17
18           System.out.println("The total count is, " + TestClass.count);
19       }
20   }
```

Output:

```
New Class Created
New Class Created
New Class Created
New Class Created
New Class Created
New Class Created
The total count is, 4
PS C:\60003220216Sujal\Expriment6>
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Object Oriented Programming using Java Laboratory (DJS22FEL22)

b. WAP to display area of square and rectangle using the concept of overloaded constructor (use parameterized, non-parameterized and copy constructor).

Code:

```java
J Expriment6B.java > class Shape > Shape()
1    class Shape {
2        private int mLength;
3        private int mBreadth;
4
5        Shape(){
6            this.mLength = 0;
7            this.mBreadth = 0;
8        }
9
10       Shape(Shape c) {
11           this.mLength = c.mLength;
12           this.mBreadth = c.mBreadth;
13       }
14
15       Shape(int size){
16           this.mLength = size;
17           this.mBreadth = size;
18       }
19
20       Shape(int l, int b){
21           this.mLength = l;
22           this.mBreadth = b;
23       }
24
25       public int getArea() {
26           return this.mLength * this.mBreadth;
27       }
28   }
29
32 ~ public static void main(String[] args) {
33       Shape a = new Shape();
34       Shape b = new Shape(size: 5);
35       Shape c = new Shape(b);
36       Shape d = new Shape(l: 3, b: 4);
37
38       System.out.println(x: "Area are: ");
39       System.out.println(a.getArea());
40       System.out.println(b.getArea());
41       System.out.println(c.getArea());
42       System.out.println(d.getArea());
43   }
44 }
45
```

Output:

```
Area are:
0
25
25
12
```