Lab 2: CONCEPTS OF CLASSES AND OBJECTS AND CONSTRUCTORS.

Objectives:

- To be familiar with objects and classes.
- To understand the concepts of constructors.
- To get the knowledge of method overloading and constructor overloading.

To create class:

```
class ClassName {
  // fields
  // methods
}
```

To create objects:

```
className object = new className();
```

Read input from keyboard:

- **Using java.util.Scanner class**

```
import java.util.Scanner;

//create object of Scanner class

Scanner sc=new Scanner(System.in);

int a=sc.nextInt();

String b=sc.next();
```

1. Write a program to take the input from the user and display the name, age,salary of an employee.
2. In this class the member functions are setdata and displaydata.
3. A program to create a class Student1 with data as name, address and age along with method printData to display the data. Create the two objects s1 , s2 to declare and access the values.

```
class Student1
{
String name, address;
int age;

void printData()
{
System.out.println("Student name = "+name);
System.out.println("Student address = "+address);
System.out.println("Student age = "+age);
}
}
public class Test
{
public static void main(String args[])
{
Student1 s1=new Student1();
Student1 s2=new Student1();
s1.name= "Ram";
s1.address= "pkr";
s1.age= 22;
s2.name= "Sita";
s2.address= "Ktm";
s2.age=23;
s1.printData();
s2.printData();
}
}
```

2. Write a program to create a class Room along with two methods getData() to get the value through argument and printData() to display the data in printData. Create the two objects s1 ,s2 to declare and access the values from class STtest.

3. Program to illustrate default and parameterized constructor.

```
class Student // creating a class
 {
        int id;
        String name;
        Student()   // creating default constructor
        {
        System.out.println("Default constructor called!!");
        }

        //creating a parameterized constructor

        Student(int i,String n)
        {
        id = i;
        name = n;
        System.out.println("Parameterized constructor   called!!");
        }

        //method to display the values

        void display()
        {
        System.out.println(id+" "+name);
```

```
    }
    public static void main(String args[])

    {

    Student s1 = new Student(); //creating objects and passing values

    s1.display(); //calling method to display the values of object

    Student s2 = new Student(1,"Ram");

    s2.display();

    Student s3 = new Student(2,"Shyam");
    s3.display();
    }
}
```

Output :

```
Default constructor called!!
0 null
Parameterized constructor called!!
1 Ram
Parameterized constructor called!!
2 Shyam
```

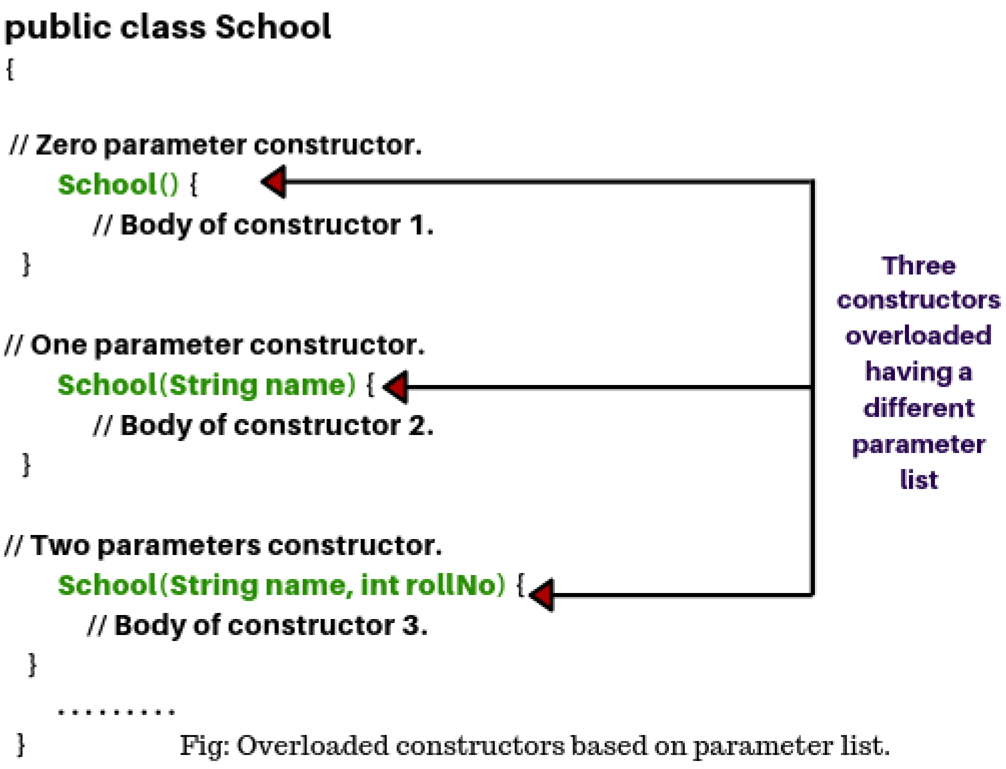
4. Write a program to show the employee id, name, salary of the employee illustrating the concept of constructor.

5. Write a program to calculate the area of a rectangle using the concept of constructor.

Constructor overloading:

- Overloaded constructors essentially have the same name (exact name of the class) and differ by number and type of arguments.

- A constructor is called depending upon the number and type of arguments passed.
- While creating the object, arguments must be passed to let the compiler know which constructor needs to be called.



```
public class School
{

 // Zero parameter constructor.
      School() {
          // Body of constructor 1.
   }

// One parameter constructor.
      School(String name) {
          // Body of constructor 2.
   }

// Two parameters constructor.
      School(String name, int rollNo) {
         // Body of constructor 3.
    }
      .........
  }
```

Three constructors overloaded having a different parameter list

Fig: Overloaded constructors based on parameter list.

6. Program to illustrate the concept of constructor overloading.

```
class Box
{
double width,height, depth;
Box(double w, double h, double d)
{
   width=w;
   height=h;
   depth=d;
}
Box()
{
width= 4;
height=5;
depth=5.5;
}
Box(double len)
{
   width=height=depth=0;
}
double volume()
{
   return(width*height*depth);
}
 public static void main(String args[])
 {
   Box b=new Box();
   double vol;
   vol=b.volume();
   System.out.println(" Volume of box b is:" +vol);
   Box b1=new Box(1,2,3);
   vol=b1.volume();
   System.out.println("Volume of box b1 is:" +vol);
}
}
```

Output:

```
 Volume of box b is:110.0
 Volume of box b1 is:6.0
```

7. Write a program to illustrate the concept of constructor overloading of the person taking no argument,1 argument and 2 argument.

Method Overloading :

- Defining methods with the same name.
- The same named method is separated by:
  - Number of arguments
  - Types of arguments.

7. Program to illustrate the concept of method overloading .

```
public class Sum {
   public int sum(int x,int y)
   {
   return(x+y);
   }
   public int sum (int x,int y,int z)
   {
   return(x+y+z);
   }
   public double sum(double x,double y,double z)
   {

   return(x+y+z);
   }
   public static void main(String args[])
   {
          Sum s=new Sum();
          System.out.println("The sum of three numbers is:"  +s.sum(10.2,20,30));
          System.out.println("The sum of two numbers is:" +s.sum(10,20));
   }
}
```

Output:

```
The sum of three numbers is:60.2
The sum of two numbers is:30
```

8. Write a program to calculate volume of cube and cuboid using the concept of method overloading.