1. Write a JAVA program to implement class mechanism. Create a class, methods and invoke them inside main method.

```
class Student
{
    String name;
    String rno;
    int age;

    Student(String rno, String name, int age)
    {
        this.rno = rno;
        this.name = name;
        this.age = age;
    }
    void displayInfo()
    {
        System.out.println("rno: " + rno);
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
public class Main
{
    public static void main(String[] args)
    {
        Student s = new Student("23A81A0601","Vasavi", 30);
        s.displayInfo();
    }
}
```

Output

```
rno: 23A81A0601
Name: Vasavi
Age: 30
```

2. Develop a java program that implements method overloading.

```
class MethodOverloading
{
        void sum (int a, int b)
        {
                int sum;
                sum = a+b;
                System.out.println(“method sum declared with two integer arguments”);
                System.out.println(“sum is “+sum);
        }
        void sum (int a, int b, int c)
        {
                int sum;
                sum = a+b+c;
                System.out.println(“this method differs by number of arguments”);
                System.out.println(“sum is “+sum);
        }
        void sum (int a, float b)
        {
                float sum;
                sum = a+b;
                System.out.println(“this method differs by type of arguments”);
                System.out.println(“sum is “+sum);
        }
        void sum (float b, int a)
        {
                float sum;
                sum = a+b;
                System.out.println(“this method differs by order of arguments”);
                System.out.println(“sum is “+sum);
        }
        public static void main (String args[])
        {
                MethodOverloading obj = new MethodOverloading();
                obj.sum(2, 4);
                obj.sum(2, 4, 6);
                obj.sum(2, 3.4f);
                obj.sum(3.6f, 2);
        }
}
```

OUTPUT:

```
        method sum declared with two integer arguments
        sum is 6
        this method differs by number of arguments
        sum is 12
        this method differs by type of arguments
        sum is 5.4
        this method differs by order of arguments   sum is 5.6
```

3. Develop a java program that implements Constructor overloading.

```
class Demo
{
        int  value1;
        int  value2;
        Demo()
        {
                value1 = 10;
                value2 = 20;
                System.out.println("Inside 1st Constructor");
        }
        Demo(int a)
        {
                 value1 = a;
                System.out.println("Inside 2nd Constructor");
        }
        Demo(int a,int b)
        {
                value1 = a;
                value2 = b;
                System.out.println("Inside 3rd Constructor");
        }
        public void display()
        {
                System.out.println("Value1 === "+value1);
                System.out.println("Value2 === "+value2);
        }
        public static void main(String args[])
        {
                Demo d1 = new Demo();
                Demo d2 = new Demo(30);
                Demo d3 = new Demo(30,40);
                d1.display();
                d2.display();
                d3.display();
        }
}
```

Output

```
Inside 1st Constructor
Inside 2nd Constructor
Inside 3rd Constructor
Value1 === 10
Value2 === 20
Value1 === 30
Value2 === 0
Value1 === 30
Value2 === 40
```

4. Write a JAVA program to implement constructor overloading.

```
public class Students
{
    int id;
    String name;
    Students()
    {
        System.out.println("this a default constructor");
    }
    Students(int i, String n)
    {
        id = i;
        name = n;
    }
    public static void main(String[] args)
    {
        Students s = new Students();
        System.out.println("\nDefault Constructor values: \n");
        System.out.println("Student Id : "+s.id + "\nStudent Name : "+s.name);
       System.out.println("\nParameterized Constructor values: \n");
        Students students = new Students(10, "David");
        System.out.println("Student Id : "+students.id + "\nStudent Name : "+students.name);
        }
    }
```

Output

```
this a default constructor
Default Constructor values:
Student Id : 0
Student Name : null
Parameterized Constructor values:
Student Id : 10
Student Name : David
```