NAME: Priyansh Salian

Batch: C-31

Roll No. : 2003148

EXPERIMENT NO. 4

Aim:

Programs on classes and objects.

Theory:

Classes And Objects:

In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other bicycles in existence, all of the same make and model. Each bicycle was built from the same set of blueprints and therefore contains the same components. In object-oriented terms, we say that your bicycle is an instance of the class of objects known as bicycles. A class is the blueprint from which individual objects are created. Instances of the class Class represent classes and interfaces in a running Java application. An enum is a kind of class and an annotation is a kind of interface. Every array also belongs to a class that is reflected as a Class object that is shared by all arrays with the same element type and number of dimensions. The primitive Java types (boolean, byte, char, short, int, long, float, and double), and the keyword void are also represented as Class objects.

Class has no public constructor. Instead Class objects are constructed automatically by the Java Virtual Machine as classes are loaded and by calls to the defineClass method in the class loader.

Calling class within main:

A class is in scope of its own members, so you can create instances of it. Take for instance a myMethod() method; it would need to create an instance of another object of its own type.

To call the method of a particular class you need to declare an object of that class and then u can call methods using the instance of object.Example: Inside main , call the myMethod() method:

```
public class Main { static void myMethod()

{ System.out.println("I just got executed!");

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

myMethod(); }
```

Following are the ways to initialize an object

1. Using Deserialization

```
ObjectInputStream objectInputStream = new ObjectInputStream(inputStream );

Tester tester5 = (MyObject) objectInputStream.readObject();
```

2. Using Constructor.forName() method

```
Tester tester4 = Tester.class.getConstructor().newInstance();
```

3. Using clone method.

```
Tester tester3 = tester1.clone();
```

4. Using Class.forName() method

```
Tester tester2 = (Tester)Class.forName("Tester").newInstance();
```

5. Using new keyword.

```
Tester tester1 = new Tester();
```

PROGRAMS:

A. WAP to read and display details of the employee using single class and its object.

Program:

```
import java.util.Scanner;
class employeeDetails{
    String name;
    int id;
    String dept;
    int salary;
    public void userInput(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter employee details:");
        System.out.print("Employee Name: ");
        name = s.next();
        System.out.print("Employee ID: ");
        id = s.nextInt();
        System.out.print("Employee Department: ");
        dept = s.next();
        System.out.print("Employee Salary: ");
        salary = s.nextInt();

    }
    public void displayDetails(){
        System.out.println("Employee Details:");
        System.out.println("Employee Name: "+name);
        System.out.println("Employee ID: "+id);
        System.out.println("Employee Department: "+dept);
        System.out.println("Employee Salary: "+salary);
    }
}
public class Employee {
    public static void main(String[] args) {
        employeeDetails e = new employeeDetails();
        e.userInput();
```

```
        e.displayDetails();

    }
}
```

Output:

```
C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>java Employee
Enter employee details:
Employee Name: Priyansh
Employee ID: 2003148
Employee Department: Computer
Employee Salary: 159800
Employee Details:
Employee Name: Priyansh
Employee ID: 2003148
Employee Department: Computer
Employee Salary: 159800

C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>
```

B. WAP to find maximum of three numbers using conditional operator, using two classes and function returning result.

Program:

```
import java.util.Scanner;
```

```
class result{
    int a,b,c;
    public void enterNum(){
        System.out.println("Enter any three numbers to be compared:");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    public int largestNum(){
        if(a>b && b>=c){
            return a;
        }
        else if(b>a && a>=c){
            return b;
        }
        else{
            return c;
        }
    }
}
public class MaxUsingClass {
    public static void main(String[] args) {
        result r = new result();
        r.enterNum();
        int max = r.largestNum();
        System.out.println("Maximum of the numbers entered is: "+max);

    }
}
```

<u>Output:</u>

```
C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>javac MaxUsingClass.java

C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>java MaxUsingClass
Enter any three numbers to be compared:
34
140
30
Maximum of the numbers entered is: 140

C:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>
```