ASSIGNMENT 06

1) Write a Java program to implement the Method Overloading concept.

```
class Test{
    void add(int no1, int no2){
        int res;
        res = no1 + no2;
        System.out.println("Sum is "+res);
    }
    void add(float value1, float value2){
        float res;
        res = value1 + value2;
        System.out.println("Sum is "+res);
    }
    void add(int no1, int no2, int no3){
        int res;
        res = no1 + no2;
        System.out.println("Sum is "+res);
    }
    void add(float value){
        float res;
        res = value + 5;
        System.out.println("Sum is "+res);
    }
}
public class Demo {
    public static void main(String[] args) {
        Test ob = new Test();
        ob.add(5,6);
        ob.add(5.0f,6.0f);
        ob.add(10,20,30);
        ob.add(5.0f);
    }
}
```

OUTPUT:

```
Sum is 11
Sum is 11.0
Sum is 30
Sum is 10.0
```

2) Design the following class hierarchies with appropriate driver class and main() method. You may add more member functions if required. Also add the parameterized constructor.

Employee

String empName

int empID

double basicSalary

static int count
double DA
double HRA
Employee()
double grossSalary()
protected void finalize()
void empDetails()

Manager
double bonus
Manager()//All constructor
void empDetails()

```
class Employee{
    String empName;
    int empID;
    double basicSalary;
    static int count;
    double DA;
    double HRA;

    Employee(){

    }
    Employee(String n, int i, double bs, int c){
        empName = n;
        empID = i;
        basicSalary = bs;
        count = c;
    }
    double grossSalary(){
        DA = basicSalary*0.15;
        HRA = basicSalary*0.6;
        return basicSalary+DA+HRA;
    }
    protected void finalize(){

    }
    void empDetails(){
        System.out.println("Name: "+empName);
        System.out.println("ID: "+empID);
        System.out.println("Basic Salary: "+basicSalary);
    }
}
class Manager extends Employee{
    double bonus;
    Manager(String n, int i, double bs, int b){
        empName = n;
        empID = i;
        basicSalary = bs;
        bonus = b;
```

```
    }
    double grossSalary(){
        DA = basicSalary*0.15;
        HRA = basicSalary*0.6;
        return basicSalary+DA+HRA+bonus;
    }
    void empDetails(){
        System.out.println("Name: "+empName);
        System.out.println("ID: "+empID);
        System.out.println("Basic Salary: "+basicSalary);
    }
}
public class Driver15 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Priyanshu", 123, 12000.00, 1);
        double gs = ob.grossSalary();
        ob.empDetails();
        System.out.println("Gross Salary: "+gs);
    }
}
```

OUTPUT:

```
Name: Priyanshu
ID: 123
Basic Salary: 12000.0
Gross Salary: 21001.0
```

3) Use the concept of abstract class and abstract method for the above class. Create an object reference of Employee class using which access the necessary member function of Manager class.

```
abstract class Employee {
    String Name;
    int number;

    Employee(String n, int i) {
        Name = n;
        number = i;
    }
    abstract void printData();
}

class Manager extends Employee {
    String title, name;
    double clubDues;
    Manager(String t, double cd, String n, int i){
        super(n, i);
        title = t;
        clubDues = cd;
    }
    void printData(){
```

```
        System.out.println(Name+" "+number+" "+title+" "+clubDues);
    }
}
public class Driver10 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Mr",2345,"Priyanshu",45679);
        ob.printData();
    }
}
```

OUTPUT:

```
Priyanshu 45679 Mr 2345.0
```

4) Design a interface Shape and implement in Square, Rectangle and Triangle class from Shape class. By creating the object reference of Shape to access the member area() of different shape.

```
import java.util.*;
interface Shape{
    void input();
    public abstract void area();
    abstract void show();
}
class Square implements Shape{
    int side, result;
    public void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the side of the square");
        side = in.nextInt();
    }
    public void area(){
        result = side * side;
    }
    public void show(){
        System.out.println("Area of square is "+result);
    }
}
class Rectangle implements Shape{
    int side1, side2, result;
    public void input(){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the side of the rectangle");
        side1 = in.nextInt();
        side2 = in.nextInt();
    }
    public void area(){
        result = side1 * side2;
    }
    public void show(){
        System.out.println("Area of Rectangle is "+result);
    }
}
class Triangle implements Shape{
```

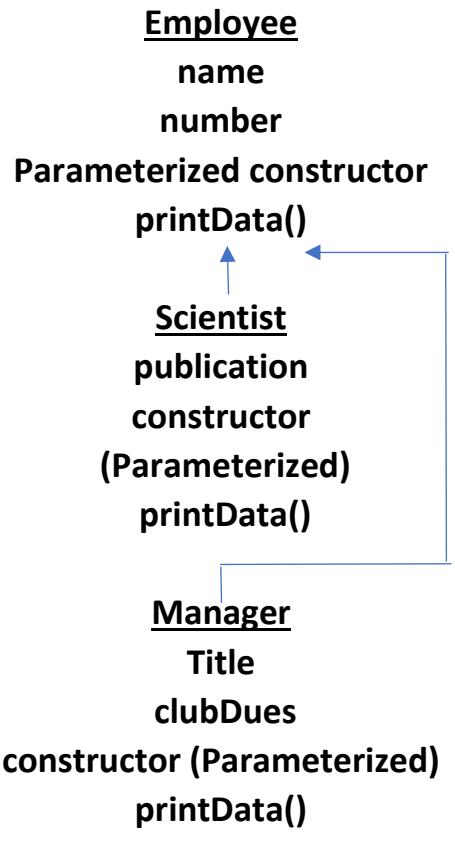
```
    double side1, side2, side3, result;
    public void input(){
        System.out.println("FOR TRIANGLE");
        System.out.println();
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the three sides of the triangle");
        side1 = in.nextDouble();
        side2 = in.nextDouble();
        side3 = in.nextDouble();
    }
    public void area(){
        double s;
        s = (side1+side2+side3)/2;
        result = Math.sqrt(s*(s-side1)*(s-side2)*(s-side3));
    }
    public void show(){
        System.out.println("Area of square is "+result);
    }
}
class DemoInterface{
    public static void main(String args[]){
        Shape ob;
        ob = new Square();
        ob.input();
        ob.area();
        ob.show();
        ob = new Rectangle();
        ob.input();
        ob.area();
        ob.show();
        ob = new Triangle();
        ob.input();
        ob.area();
        ob.show();
    }
}
```

OUTPUT:

```
Enter the side of the square
2
Area of square is 4
Enter the side of the rectangle
2 4
Area of Rectangle is 8
FOR TRIANGLE

Enter the three sides of the triangle
4 2 6
Area of square is 0.0
```

5) Design following class hierarchies with appropriate main() function. You may add more member functions if required.




```
abstract class Employee {
    String Name;
    int number;

    Employee(String n, int i) {
        Name = n;
        number = i;
    }
    abstract void printData();
}

class Manager extends Employee {
    String title, name;
    double clubDues;
    Manager(String t, double cd, String n, int i){
        super(n, i);
        title = t;
        clubDues = cd;
    }
    void printData(){
        System.out.println(Name+" "+number+" "+title+" "+clubDues);
    }
}
class Scientist extends Employee {
    String publication, name;
    Scientist(String p, String n, int i){
        super(n, i);
        publication = p;
    }
    void printData(){
        System.out.println(Name+" "+number+" "+publication);
    }
```

```
}
public class Driver10 {
    public static void main(String[] args) {
        Employee ob;
        ob = new Manager("Mr",2345,"Priyanshu",45679);
        ob.printData();
        ob = new Scientist("KP","Chandan",23456);
        ob.printData();
    }
}
```

OUTPUT:

```
Priyanshu 45679 Mr 2345.0
Chandan 23456 KP
```

NAME – PRIYANSHU MALLICK
SIC – 21BCSF11
ROLL. NO – 30
SEC – B, GROUP – B2
BRANCH - CSE