ASSIGNMENT-4

Q.1. Write a java program which initialization earning of an employee. The program should calculate the income tax to be paid by the employee as per the criteria given below:

Slab rate	IT rate
Upto Rs. 45,000	Nil
Upto Rs. 55,000	10% on additional amount
Upto Rs. 1,00,000	20% on additional amount
Above Rs. 1,50,000	30% on additional amount

- Code:

```
class emp{
    int sal,id,tax;
    String post,name;
    emp(int id, String name,int sal, String post){
        this.id = id;
        this.sal = sal;
        this.name = name;
        this.post = post;
        if (sal <= 45000){
            tax = 0;
        } else if (sal<= 99999 || sal >= 55000) {
            tax = (sal*10)/100;
        } else if (sal <= 14999 ||sal >= 100000) {
            tax = (sal*20)/100;
        } else if (sal >= 150000) {
            tax = (sal*30)/100;
        }
    }

    public int getTax() {
        return tax;
    }
}
public class assignment4_1 {
    public static void main(String[] args) {
        emp junior_programmer = new emp(101,"Ravi",40000, "Junior
Programmer");
        emp senior_programmer = new emp(102,"Raj",60000, "Senior
Programmer");
        emp HR = new emp(103,"Sunny",120000, "H.R.");
        emp Manager = new emp(104,"Pradeep",160000, "Manager");


        System.out.println("Tax Pay By " + junior_programmer.name
+ " Is: " + junior_programmer.getTax());
        System.out.println("Tax Pay By " + senior_programmer.name
+ " Is: " + senior_programmer.getTax());
        System.out.println("Tax Pay By " + HR.name + " Is: " +
HR.getTax());
        System.out.println("Tax Pay By " + Manager.name + " Is: "
+ Manager.getTax());
    }
}
```

- **Output:**

```
Tax Pay By Ravi Is: 0
Tax Pay By Raj Is: 6000
Tax Pay By Sunny Is: 12000
Tax Pay By Pradeep Is: 16000
```

ASSIGNMENT-4

Q.2. Create a abstract class employee, having its properties & abstract function for calculating net salary and displaying the information. Drive manager & clerk class from this abstract class & implement the abstract method net salary and override the display method.

- Code:

```
abstract class employee {
    protected String name;
    protected double basicSalary;
    public employee(String name, double basicSalary) {
        this.name = name;
        this.basicSalary = basicSalary;
    }
    public abstract double calculateNetSalary();
    public abstract void displayInformation();
}
class manager extends employee {
    private double bonus;
    public manager(String name, double basicSalary, double bonus) 
{
        super(name, basicSalary);
        this.bonus = bonus;
    }
    @Override
    public double calculateNetSalary() {
        return basicSalary + bonus;
    }
    @Override
    public void displayInformation() {
        System.out.println("Manager: " + name);
        System.out.println("Basic Salary: " + basicSalary);
        System.out.println("Bonus: " + bonus);
        System.out.println("Net Salary: " + calculateNetSalary());
    }
}
class Clerk extends employee {
    private double overtimeHours;
    public Clerk(String name, double basicSalary, double
overtimeHours) {
        super(name, basicSalary);
        this.overtimeHours = overtimeHours;
    }
    @Override
    public double calculateNetSalary() {
        return basicSalary + (overtimeHours * 10);
    }
    @Override
    public void displayInformation() {
        System.out.println("Clerk: " + name);
        System.out.println("Basic Salary: " + basicSalary);
        System.out.println("Overtime Hours: " + overtimeHours);
        System.out.println("Net Salary: " + calculateNetSalary());
    }
}
public class assignment4_2 {
    public static void main(String[] args) {
        manager manager = new manager("Emp1", 50000, 10000);
        Clerk clerk = new Clerk("Emp2", 30000, 20);
        manager.displayInformation();
        System.out.println("--------------------");
        clerk.displayInformation();
    }
}
```

ASSIGNMENT-4

- **Output:**

```
Manager: Emp1

Basic Salary: 50000.0

Bonus: 10000.0

Net Salary: 60000.0

--------------------

Clerk: Emp2

Basic Salary: 30000.0

Overtime Hours: 20.0

Net Salary: 30200.0
```

ASSIGNMENT-4

Q.3. Write a Java program to create an interface Shape with the getArea() method.Create three classes Rectangle, Square, and Triangle that implement the Shape interface. Implement the getArea() method for each of the three classes.

- Code:

```
interface shape {
    double getArea();
}
class Rectangle implements shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    @Override
    public double getArea() {
        return length * width;
    }
}
class Square implements shape {
    private double side;
    public Square(double side) {
        this.side = side;
    }
    @Override
    public double getArea() {
        return side * side;
    }
}
class Triangle implements shape {
    private double base;
    private double height;
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    @Override
    public double getArea() {
        return 0.5 * base * height;
    }
}
public class assignment4_3 {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 8);
        Square square = new Square(4);
        Triangle triangle = new Triangle(6, 10);
        System.out.println("Area of Rectangle: " +
rectangle.getArea());
        System.out.println("Area of Square: " + square.getArea());
        System.out.println("Area of Triangle: " +
triangle.getArea());
    }
}
```

- **Output:**

```
Area of Rectangle: 40.0
Area of Square: 16.0
Area of Triangle: 30.0
```