

**Q1)**

```
public abstract class DiscountPolicy {
    public abstract double computeDiscount(int count, double itemCost);
}

public class BulkDiscount extends DiscountPolicy {
    private int minimum;
    private double rate;

    public BulkDiscount(int m, double r) {
        rate = r;
        minimum = m;
    }

    public double computeDiscount(int a, double b) {
        if (a > minimum)
            return b*rate/100;
        else
            return 0;
    }
}

public class OtherDiscount extends DiscountPolicy {
    public double computeDiscount(int a, double b) {
        if (a > 8)
            return 0.3*b;
        else if (a > 5)
            return 0.2*b;
        else if (a > 2)
            return 0.1*b;
        else
            return 0;
    }
}
```

**Q2)**

```
public interface Interest {  
    public double computeInterest();  
}
```

```
public class SavingAccount implements Interest { //In interface, you IMPLEMENT  
    private double balance;  
    public SavingAccount(double a) { balance = a; }
```

```
    public double computeInterest() {  
        return balance * 0.005 / 12;  
    }
```

```
    public void display() {  
        System.out.println("Saving Account Balance: " + balance);  
        System.out.println("This month interest: " + computeInterest());  
    }  
}
```

```
public class FixedAccount implements Interest {  
    private double balance;  
    public FixedAccount(double a) { balance = a; }
```

```
    public double computeInterest() {  
        return balance * 0.03 / 12;  
    }
```

```
    public void display() {  
        System.out.println("Fixed Account Balance: " + balance);  
        System.out.println("This month interest: " + computeInterest());  
    }  
}
```

```
    public static void main(String[] args) {  
        SavingAccount a = new SavingAccount(12000);
```

```
FixedAccount b = new FixedAccount(5000);
a.display();
b.display();
}
```

### Q3)

// Abstract class Shape

```
abstract class Shape {
    protected double area;
```

// Getter for area

```
public double getArea() {
    return area;
}
```

// Setter for area

```
public void setArea(double area) {
    this.area = area;
}
```

// Abstract methods

```
public abstract void RectangleArea(double length, double breadth);
public abstract void SquareArea(double side);
public abstract void CircleArea(double radius);
}
```

// Concrete class Area extending Shape

```
class Area extends Shape {
```

@Override

```
public void RectangleArea(double length, double breadth) {
    area = length * breadth;
}
```

@Override

```
public void SquareArea(double side) {
    area = side * side;
}
```

```

@Override
public void CircleArea(double radius) {
    area = Math.PI * radius * radius;
}
}

```

// Main class to test the implementation

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

```
public class TestShape {
```

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

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Array to store shape objects
```

```
        Shape[] shapes = new Shape[3]; //this array is to store the objects of each of the classes.
```

```
        shapes[0] = new Area(); // Rectangle
```

```
        shapes[1] = new Area(); // Square
```

```
        shapes[2] = new Area(); // Circle
```

```
        // Display options to the user
```

```
        System.out.println("Press (1) for calculating Rectangle Area");
```

```
        System.out.println("Press (2) for calculating Square Area");
```

```
        System.out.println("Press (3) for calculating Circle Area");
```

```
        // Get user choice
```

```
        int choice = scanner.nextInt();
```

```
        switch (choice) {
```

```
            case 1:
```

```
                System.out.println("Enter length and breadth of the rectangle:");
```

```
                double length = scanner.nextDouble();
```

```
                double breadth = scanner.nextDouble();
```

```
                shapes[0].RectangleArea(length, breadth);
```

```
                System.out.println("Rectangle Area: " + shapes[0].getArea());
```

```
                break;
```

```
            case 2:
```

```
                System.out.println("Enter the side of the square:");
```

```
                double side = scanner.nextDouble();
```

```
                shapes[1].SquareArea(side);
```

```
                System.out.println("Square Area: " + shapes[1].getArea());
```

```
                break;
```

```
case 3:
    System.out.println("Enter the radius of the circle:");
    double radius = scanner.nextDouble();
    shapes[2].CircleArea(radius);
    System.out.println("Circle Area: " + shapes[2].getArea());
    break;
default:
    System.out.println("Invalid choice");
}
}
}
```