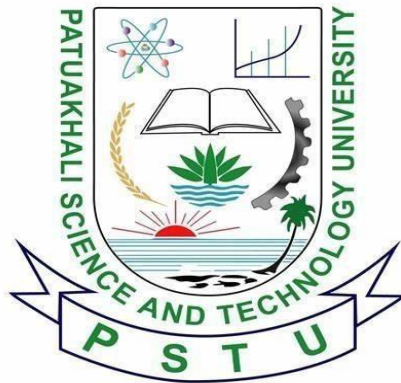


PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY



Course Code: CIT-121

SUBMITTED TO:

Lecturer Sarna Majumder Mam
Department of Computer and Communication
Engineering
Faculty of Computer Science And Engineering

SUBMITTED BY:

Name: MD Noushad Bhuiyan
ID: 2102038, Registration No: 10165
Faculty of Computer Science and Engineering

Date of submission: 03-1-2024

Polymorphism

Problem 1.

```
public class Shape{
    public static void show()
    {
        System.out.println("This is a shape");
    }
    public static void getInfo()
    {
        System.out.println("this is Shape class");
    }
}

public class Circle extends Shape{
    public static void show()
    {
        System.out.println("This is a Circle");
    }
}

public class Rectangle extends Shape{
    public static void show()
    {
        System.out.println("This is a Rectangle");
    }
}

public class main{
    public static void main(String[] args)
    {
        Shape shape = new Shape();
        Circle circle = new Circle();
        Rectangle rect = new Rectangle();
        System.out.println("This should show base class show method");
        shape.show();
        System.out.println("This should show circle class show method");
        circle.show();
        System.out.println("This should show rectangle class show method");
        rect.show();
        System.out.println("This should show base class getinfo method");
        circle.getInfo();
        System.out.println("This should show base class getinfo method");
        rect.getInfo();
    }
}
```

Problem 02

```
public class SuperClass {
    protected int data1;

    public SuperClass(int initialData1) {
        this.data1 = initialData1;
    }

    public int getData1() {
        return data1;
    }

    public void setData1(int newData1) {
        this.data1 = newData1;
    }
}

public class SubClass extends SuperClass {
    private int data2;

    public SubClass(int initialData1, int initialData2) {
        super(initialData1);
        this.data2 = initialData2;
    }

    public int getData2() {
        return data2;
    }

    public void setData2(int newData2) {
        this.data2 = newData2;
    }

    public String checkCondition() {
        if (data1 == 10 && data2 == 15) {
            return "Condition True!";
        } else {
            return "Condition False!";
        }
    }
}

public class main{
    public static void main(String[] args){
        SubClass test1 = new SubClass(12,34);
        System.out.println("For 12 and 34 : "+test1.checkCondition());
        SubClass test2 = new SubClass(10,15);
    }
}
```

```
        System.out.println("For 10 and 15 : "+test2.checkCondition());
    }
}
```

Problem 03

```
public class Pizza{
    private double size;
    private double Cheese_toppings;
    private double Papperoni_toppings;
    private double Ham_toppings;
    public Pizza(double d1,double d2,double d3,double d4)
    {
        this.size=d1;
        this.Cheese_toppings=d2;
        this.Papperoni_toppings=d3;
        this.Ham_toppings=d4;
    }
    public void setSize(double size)
    {
        this.size = size;
    }
    public double getSize()
    {
        return size;
    }
    //
    public void setCheese_toppings(double Cheese_toppings)
    {
        this.Cheese_toppings = Cheese_toppings;
    }
    public double getCheese_toppings()
    {
        return Cheese_toppings;
    }
    //
    public void setPapperoni_toppings(double Papperoni_toppings)
    {
        this.Papperoni_toppings = Papperoni_toppings;
    }
    public double getPapperoni_toppings()
    {
        return Papperoni_toppings;
    }
    //
}
```

```

public void setHam_toppings(double Ham_toppings)
{
    this.Ham_toppings = Ham_toppings;
}
public double getHam_toppings()
{
    return Ham_toppings;
}
public double CalcCost()
{
    double cost =0;
    double sizeCost=0;
    if(size==1)
        sizeCost=10;
    if(size==2)
        sizeCost=12;
    if(size==3)
        sizeCost=14;
    cost=sizeCost+(Cheese_toppings+Ham_toppings+Papperoni_toppings)*2;

    System.out.println("Total cost of the pizza is: ");
    return cost;
}
public void getDescription()
{
    if(size==1)
    {
        System.out.println("The size of the pizza is small.\n there are
"+Cheese_toppings+" Cheese toppings, \n"+Papperoni_toppings+" Papperoni toppings \nand
"+Ham_toppings+ " Ham toppings");
    }
    if(size==2)
    {
        System.out.println("The size of the pizza is medium.\n there are
"+Cheese_toppings+" Cheese toppings, \n"+Papperoni_toppings+" Papperoni toppings \nand
"+Ham_toppings+ " Ham toppings");
    }
    if(size==3)
    {
        System.out.println("The size of the pizza is large.\n there are
"+Cheese_toppings+" Cheese toppings, \n"+Papperoni_toppings+" Papperoni toppings \nand
"+Ham_toppings+ " Ham toppings");
    }
}
}
}

```

```

import java.util.Scanner;
public class main{
    public static void main(String[] args)
    {
        double sum=0;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the number of Pizza you want to order:");
        int n=s.nextInt();
        int a[]=new int[n];

        for(int i=0; i<n; i++)
        {
            System.out.println("Enter the size of pizza:\nPress 1 for small pizza\nPress 2
for medium pizza\nPress 3 for large pizza");
            double size=s.nextDouble();
            System.out.println("Enter the number of Cheese toppings: ");
            double ct=s.nextDouble();
            System.out.println("Enter the number of Papperoni toppings: ");
            double pt=s.nextDouble();
            System.out.println("Enter the number of Ham toppings: ");
            double ht=s.nextDouble();
            Pizza pz=new Pizza(size,ct,pt,ht);
            a[i]=(int)pz.CalcCost();
            //System.out.println("Pizza number "+(i+1)+" prize is "+ pz.CalcCost()+"$");
            sum+=pz.CalcCost();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("Cost of pizza number "+(i+1)+" is "+a[i]+"$");
        }
        System.out.println("The total cost of "+n+" pizza's are: "+sum+"$");
    }
}

```

Problem 04

```

public abstract class Abstract{

    public Abstract(){
        System.out.println("this is a abstract class");
    }

    abstract void a_method();
}

```

```

    void method()
    {
        System.out.println("this is a normal method of abstract class");
    }
}
public class SubClass extends Abstract{
    // @Override
    void a_method()
    {
        System.out.println("this is an abstract method");
    }
}
public class Main
{
    public static void main(String[] args)
    {
        SubClass ab;
        ab = new SubClass();
        ab.method();
        ab.a_method();
    }
}

```

Problem 05

```

abstract class Animals {
    abstract void cats();
    abstract void dogs();
}

class Cats extends Animals {
    void cats() {
        System.out.println("Cats meow");
    }

    void dogs() {
        System.out.println("Cats don't bark!");
    }
}

class Dogs extends Animals {
    void cats() {
        System.out.println("Dogs don't meow!");
    }
}

```

```

    void dogs() {
        System.out.println("Dogs bark");
    }
}

public class Main {
    public static void main(String[] args) {

        Cats myCats = new Cats();

        myCats.cats();
        myCats.dogs();
        System.out.println();
        Dogs myDogs = new Dogs();

        myDogs.cats();
        myDogs.dogs();
    }
}

```

Problem 06

```

abstract class Shape {
    abstract void RectangleArea(double length, double breadth);
    abstract void SquareArea(double side);
    abstract void CircleArea(double radius);
}

class Area extends Shape {
    void RectangleArea(double length, double breadth) {
        double area = length * breadth;
        System.out.println("Area of Rectangle: " + area);
    }
    void SquareArea(double side) {
        double area = side * side;
        System.out.println("Area of Square: " + area);
    }

    void CircleArea(double radius) {
        double area = Math.PI * radius * radius;
        System.out.println("Area of Circle: " + area);
    }
}

```



```

}

public class Main {
    public static void main(String[] args) {
        Area areaCalculator = new Area();

        areaCalculator.RectangleArea(5.0, 8.0);
        areaCalculator.SquareArea(4.5);
        areaCalculator.CircleArea(3.0);
    }
}

```

Problem 07

```

abstract class Shape {
    abstract void RectangleArea(double length, double breadth);
    abstract void SquareArea(double side);
    abstract void CircleArea(double radius);
}

class Area extends Shape {
    void RectangleArea(double length, double breadth) {
        double area = length * breadth;
        System.out.println("Area of Rectangle: " + area);
    }

    void SquareArea(double side) {
        double area = side * side;
        System.out.println("Area of Square: " + area);
    }

    void CircleArea(double radius) {
        double area = Math.PI * radius * radius;
        System.out.println("Area of Circle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        Area[] rectangles = new Area[4];

        Area[] squares = new Area[4];

        Area[] circles = new Area[5];
    }
}

```

```

        for (int i = 0; i < 4; i++) {
            rectangles[i] = new Area();
            squares[i] = new Area();
        }
        for (int i = 0; i < 5; i++) {
            circles[i] = new Area();
        }

        System.out.println("Areas of Rectangles:");
        for (int i = 0; i < 4; i++) {
            rectangles[i].RectangleArea(5.0, 8.0);
        }

        System.out.println("\nAreas of Squares:");
        for (int i = 0; i < 4; i++) {
            squares[i].SquareArea(4.5);
        }

        System.out.println("\nAreas of Circles:");
        for (int i = 0; i < 5; i++) {
            circles[i].CircleArea(3.0);
        }
    }
}

```

Problem 08

```

interface TVremote {
    void turnOn();
    void turnOff();
    void changeChannel(int channel);
}

interface SmartTVremote extends TVremote {
    void connectToInternet();
    void useSmartFeatures();
}

class TV implements TVremote {
    private boolean isOn;
    private int currentChannel;

    public TV() {

```

```

        this.isOn = false;
        this.currentChannel = 0;
    }

    public void turnOn() {
        if (!isOn) {
            System.out.println("Turning the TV on");
            isOn = true;
        } else {
            System.out.println("TV is already on");
        }
    }

    public void turnOff() {
        if (isOn) {
            System.out.println("Turning the TV off");
            isOn = false;
        } else {
            System.out.println("TV is already off");
        }
    }

    public void changeChannel(int channel) {
        if (isOn) {
            System.out.println("Changing channel to " + channel);
            currentChannel = channel;
        } else {
            System.out.println("Cannot change channel, TV is off");
        }
    }
}

public class Main {
    public static void main(String[] args) {

        TV myTV = new TV();

        myTV.turnOn();
        myTV.changeChannel(5);
        myTV.turnOff();
    }
}

```