

Polymorphism

```
/*
 * 1. Write a console based program to implement polymorphism using inheritance.
 * Consider the example of Shape as base class with method show. And then a child class
 * Circle and Rectangle which inherit the base class Shape and override its method show.
 * Add one more Method with the name of getinfo0. This method would display the class name
 * in which it is implemented. Do not override this method. When you will call the method
 * getinfo0 with child object it would still show the name of the base class, which implies
 * that method has been directly inherited and was not overridden.
 */

class Shape{
    void show(){
        System.out.println("This is a shape");
    }
    void getinfo0(){
        System.out.println("This is a shape");
    }
}
class Circle extends Shape{
    @Override
    void show(){
        System.out.println("This is a circle");
    }
}
class Rectangle extends Shape{
    @Override
    void show(){
        System.out.println("This is a rectangle");
    }
}

class polymorphismByInheritance{
    public static void main(String[] args) {
        Shape objShape = new Shape();
        objShape.show();
        objShape.getinfo0();
        Shape objCircShape= new Circle();
        objCircShape.show();
        objCircShape.getinfo0();
        Shape objRectangleShape = new Rectangle();
        objRectangleShape.show();
        objRectangleShape.getinfo0();
    }
}
```

```
/*
 * 2. Write a subclass called SubClass that is derived from SuperClass and that adds
 * an integer data field called data2 and a public method called checkCondition() that will
 * check if data1 is equal to 10 and data2 is equal to 15, the checkCondition() method should
 * return "Condition True!". Also, create methods called setData2() and getData2() for setting and
 * retrieving the value of data1 and data2, as well as a constructor that accepts arguments for the
 * starting values of data1 and data2 data1 is data member of SuperClass.
 */

class superClass{
    int data1 = 10;
}

class subClass extends superClass{
    int data1;
    int data2;
    void getData1(){
        this.data1 = super.data1;
    }
    void setData2(int data2){
        this.data2 = data2;
    }
    void checkCondition(){
        System.out.println("Condition " + (data1==10 && data2==15) + "!");
    }
}

public class polymorphismWithInheritance {
    public static void main(String[] args) {
        subClass objSubClass = new subClass();
    }
}
```

```

        objSubClass.getData1();
        objSubClass.setData2(15);
        objSubClass.checkCondition();
    }
}

```

```

/*
 * 3. Create a class named Pizza that stores information about a single pizza. It should contain the following:
 * Private instance variables to store the size of the pizza (either small, medium, or large), the number of cheese toppings,
 * the number of pepperoni toppings, and the number of ham toppings Constructors) that set all of the instance variables.
 * Public methods to get and set the instance variables. A public method named calcCost() that returns a double
 * that is the cost of the pizza.
 *
 * Pizza cost is determined by:
 * Small: $10 + $2 per topping
 * Medium: $12 + $2 per topping
 * Large: $14 + $2 per topping
 *
 * • public method named getDescription() that returns a String containing the pizza size, quantity of each topping.
 * Write test code to create several pizzas and output their descriptions. For example, a large pizza with one cheese,
 * one pepperoni and two ham toppings should cost a total of $22. Now Create a PizzaOrder class that allows up to three pizzas
 * to be saved in an order. Each pizza saved should be a Pizza object. Create a method calcTotal that returns the cost of order.
 * In the runner order two pizzas and return the total cost.
 */

import java.util.Scanner;

class pizza{
    private String size;
    private int cheese;
    private int pepperoni;
    private int ham;

    pizza(String size, int cheese, int pepperoni, int ham) {
        this.size = size;
        this.cheese = cheese;
        this.pepperoni = pepperoni;
        this.ham = ham;
    }

    public void setProp(String size, int cheese, int pepperoni, int ham) {
        this.size = size;
        this.cheese = cheese;
        this.pepperoni = pepperoni;
        this.ham = ham;
    }

    public double calcCost() {
        double total = 0;
        if (size.equals("Small"))
            total += 10;
        else if (size.equals("Medium"))
            total += 12;
        else if (size.equals("Large"))
            total += 14;

        total += (2*(cheese + pepperoni + ham));

        return total;
    }

    public String getDescription() {
        return "Size: " + size + ", Cheese: " + cheese + ", Pepperoni: " + pepperoni + ", Ham: " + ham;
    }
}

public class pizzaOrder {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        pizza[] objPizza = new pizza[3];
        System.out.print("Enter the quantity: ");
        int quantity = scan.nextInt();
        if (quantity > 3) {
            System.out.println("You can only order 3 pizzas at a time.");
            System.exit(0);
        }
        else if (quantity <= 3){
            for (int i = 0; i < quantity; i++) {
                scan.nextLine();
                System.out.println("\nEnter the details of Pizza you want: ");
                System.out.print("Enter Size: ");
                String size = scan.nextLine();
            }
        }
    }
}

```

```

        System.out.print("Enter Cheese quantity: ");
        int cheese = scan.nextInt();
        System.out.print("Enter Pepperoni quantity: ");
        int pepperoni = scan.nextInt();
        System.out.print("Enter Ham quantity: ");
        int ham = scan.nextInt();

        objPizza[i] = new pizza(size, cheese, pepperoni, ham);
        objPizza[i].getDescription();
        objPizza[i].calcCost();
        System.out.println("\n" + objPizza[i].getDescription());
        System.out.println("\nThis Pizza Costs: $ " + objPizza[i].calcCost());

    }
}

double totalCost = 0;
for (int i = 0; i < quantity; i++) {
    totalCost += objPizza[i].calcCost();
}

System.out.println("\nOrder Placed! Total Cost is $ " + totalCost);

scan.close();
}
}

```