**BasicPizza Class**

public class BasicPizza implements Pizza {

@Override

public String getDescription() {

return "Pizza";

}

@Override

public double getCost() {

return 7.00; // Base cost of a basic small pizza

}

}

**CheeseDecorator Class**

// Decorator for adding extra cheese

public class CheeseDecorator extends PizzaDecorator {

public CheeseDecorator(Pizza pizza) {

super(pizza);

}

@Override

public String getDescription() {

return pizza.getDescription() + " + Extra Cheese";

}

@Override

public double getCost() {

return pizza.getCost() + 1.50;

}

}

**MushroomDecorator Class**

public class MushroomDecorator extends PizzaDecorator {

public MushroomDecorator(Pizza pizza) {

super(pizza);

}

@Override

public String getDescription() {

return pizza.getDescription() + " + Mushrooms";

}

@Override

public double getCost() {

return pizza.getCost() + 1.00; // cost for mushrooms

}

}

**Pizza Class**

public interface Pizza {

String getDescription();

double getCost();

}

**PizzaDecorator Class**

public abstract class PizzaDecorator implements Pizza {

protected Pizza pizza;

public PizzaDecorator(Pizza pizza) {

this.pizza = pizza;

}

@Override

public String getDescription() {

return pizza.getDescription();

}

@Override

public double getCost() {

return pizza.getCost();

}

}

**PizzaOrderSystem Class**

import java.util.Scanner;

public class PizzaOrderSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Pizza pizza = new BasicPizza();

System.out.println(

"Cost of a basic pizza is $7.0\n\n" +

"Select toppings to add:\n" +

" 0. None (+$0.0)\n" +

" 1. Extra Cheese (+$1.5)\n" +

" 2. Mushrooms (+$1.0)\n" +

"Enter the numbers of your choices separated by spaces (e.g., 1,2,3) or 0 for no toppings:"

);

// [TODO] Allow for more toppings for a pizza

String topping = scanner.nextLine();

switch (topping) {

case "0":

break;

case "1":

// Add extra cheese

pizza = new CheeseDecorator(pizza);

break;

case "2":

// Add a mushroom topping

pizza = new MushroomDecorator(pizza);

break;

// [TODO] Add PepperoniDecorator here (the cost is $2.00)

default:

System.out.println("Invalid choice: " + topping);

scanner.close();

return;

}

// [TODO] Prompt the user to select a pizza size (S, M, L, XL)

// Then wrap the pizza with a SizeDecorator to adjust its size and price

// Display final pizza description and cost

System.out.println("You ordered: " + pizza.getDescription());

System.out.println("Total cost: $" + pizza.getCost());

scanner.close();

}

}