*// Java program to demonstrate Decorator*

*// pattern*

*// Abstract Pizza class (All classes extend*

*// from this)*

**abstract class Pizza**

{

*// it is an abstract pizza*

    String description = "Unkknown Pizza";

    public String getDescription()

    {

        return description;

    }

    public abstract int getCost();

}

*// The decorator class :  It extends Pizza to be*

*// interchangable with it topings decorator can*

*// also be implemented as an interface*

**abstract class ToppingsDecorator extends Pizza**

{

    public abstract String getDescription();

}

*// Concrete pizza classes*

**class PeppyPaneer extends Pizza**

{

    public PeppyPaneer() { description = "PeppyPaneer"; }

    public int getCost() {  return 100; }

}

**class FarmHouse extends Pizza**

{

    public FarmHouse() {  description = "FarmHouse"; }

    public int getCost() { return 200; }

}

**class Margherita extends Pizza**

{

    public Margherita()  { description = "Margherita"; }

    public int getCost() { return 100;  }

}

**class ChickenFiesta extends Pizza**

{

    public ChickenFiesta() { description = "ChickenFiesta";}

    public int getCost() { return 200; }

}

**class SimplePizza extends Pizza**

{

public SimplePizza() { description = "SimplePizza"; }

public int getCost() {  return 50;  }

}

*// Concrete toppings classes*

**class FreshTomato extends ToppingsDecorator**

{

*// we need a reference to obj we are decorating*

    Pizza pizza;

    public FreshTomato(Pizza pizza) { this.pizza = pizza; }

    public String getDescription() {

        return pizza.getDescription() + ", Fresh Tomato ";

    }

    public int getCost() { return 40 + pizza.getCost(); }

}

**class Barbeque extends ToppingsDecorator**

{

    Pizza pizza;

    public Barbeque(Pizza pizza) {  this.pizza = pizza;  }

    public String getDescription() {

        return pizza.getDescription() + ", Barbeque ";

    }

    public int getCost() { return 90 + pizza.getCost(); }

}

**class Paneer extends ToppingsDecorator**

{

    Pizza pizza;

    public Paneer(Pizza pizza)  {  this.pizza = pizza; }

    public String getDescription() {

        return pizza.getDescription() + ", Paneer ";

    }

    public int getCost()  {  return 70 + pizza.getCost(); }

}

*// Other toppings can be coded in a similar way*

*// Driver class and method*

**class PizzaStore**

{

**public static void main(String args[])**

    {

*// create new margherita pizza*

        Pizza pizza = new Margherita();

        System.out.println( pizza.getDescription() +

                         " Cost :" + pizza.getCost());

*// create new FarmHouse pizza*

        Pizza pizza2 = new FarmHouse();

*// decorate it with freshtomato topping*

        pizza2 = new FreshTomato(pizza2);

*//decorate it with paneer topping*

        pizza2 = new Paneer(pizza2);

        System.out.println( pizza2.getDescription() +

                         " Cost :" + pizza2.getCost());

        Pizza pizza3 = new Barbeque(null);    //no specific pizza

        System.out.println( pizza3.getDescription() + "  Cost :" + pizza3.getCost());

   }

}

Margherita Cost :100

FarmHouse, Fresh Tomato , Paneer Cost :310