Repo:

https://github.com/Cloudsted/CS4800Homework5

Code

```java
public interface FoodItem {

    double cost();

}


public class Burger implements FoodItem {

    public double cost() {

        return 5.00;

    }

}


public class Fries implements FoodItem {

    public double cost() {

        return 2.50;

    }

}


public class HotDog implements FoodItem {

    public double cost() {

        return 3.00;

    }

}


public abstract class FoodItemDecorator implements FoodItem {

    protected FoodItem decoratedFoodItem;
```

```java
    public FoodItemDecorator(FoodItem decoratedFoodItem) {

        this.decoratedFoodItem = decoratedFoodItem;

    }


    public double cost() {

        return decoratedFoodItem.cost();

    }

}


public class Cheese extends FoodItemDecorator {

    public Cheese(FoodItem decoratedFoodItem) {

        super(decoratedFoodItem);

    }


    public double cost() {

        return super.cost() + 0.50;

    }

}


public class ExtraMeat extends FoodItemDecorator {

    public ExtraMeat(FoodItem decoratedFoodItem) {

        super(decoratedFoodItem);

    }


    public double cost() {

        return super.cost() + 1.00;

    }

}
```

```java
import java.util.ArrayList;

import java.util.List;


public class Order {

    private List<FoodItem> items = new ArrayList<>();


    public void addItem(FoodItem item) {

        items.add(item);

    }


    public double getTotalCost() {

        return items.stream().mapToDouble(FoodItem::cost).sum();

    }

}

public class LoyaltyDiscount {

    public static double applyDiscount(double totalCost, String loyaltyLevel) {

        switch (loyaltyLevel) {

            case "GOLD":

                return totalCost * 0.8;

            case "SILVER":

                return totalCost * 0.9;

            default:

                return totalCost;

        }

    }

}
```

Unit Testing

```java
import static org.junit.Assert.*;
import org.junit.Test;

public class RestaurantTest {
    @Test
    public void testOrderTotalCost() {
        Order order = new Order();
        order.addItem(new Burger());
        order.addItem(new Cheese(new Burger()));
        order.addItem(new ExtraMeat(new HotDog()));


        assertEquals(10.50, order.getTotalCost(), 0.001);
    }


    @Test
    public void testLoyaltyDiscount() {
        double totalCost = 100.00;
        assertEquals(80.00, LoyaltyDiscount.applyDiscount(totalCost, "GOLD"), 0.001);
        assertEquals(90.00, LoyaltyDiscount.applyDiscount(totalCost, "SILVER"), 0.001);
        assertEquals(100.00, LoyaltyDiscount.applyDiscount(totalCost, ""), 0.001);
    }
}
```