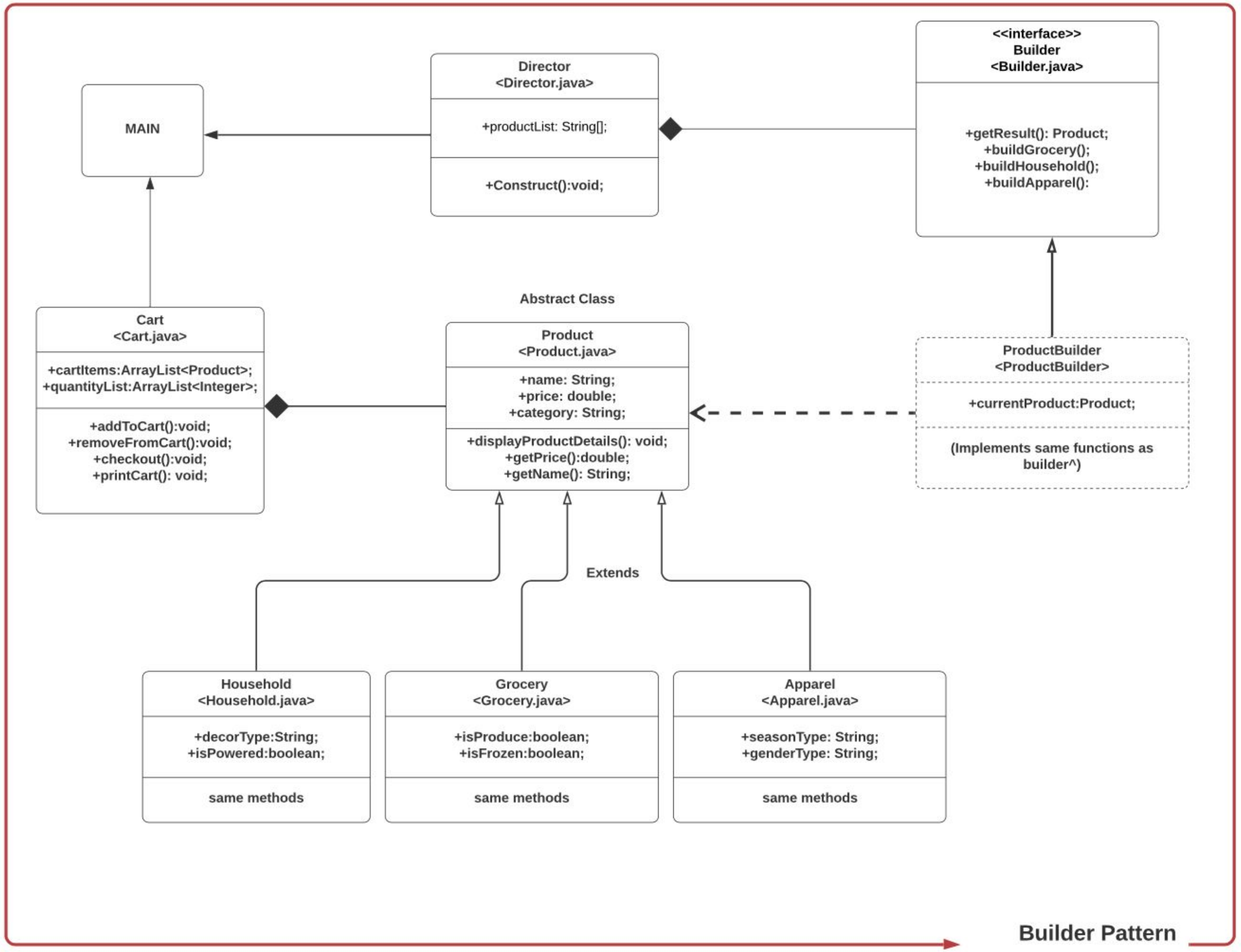
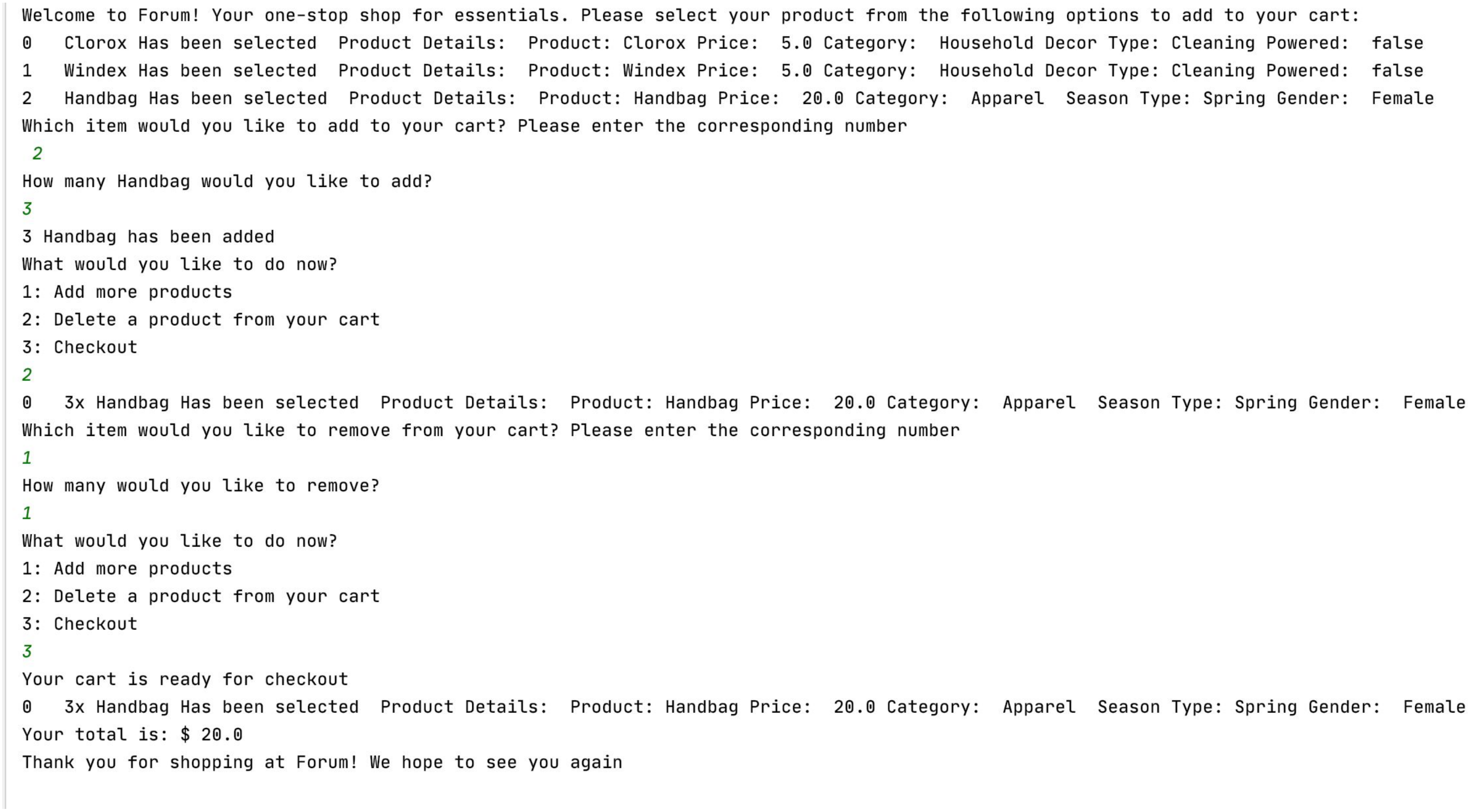
Class Project: Forum - E-commerce

Alina Akram - MET CS665 Software and Design Patterns



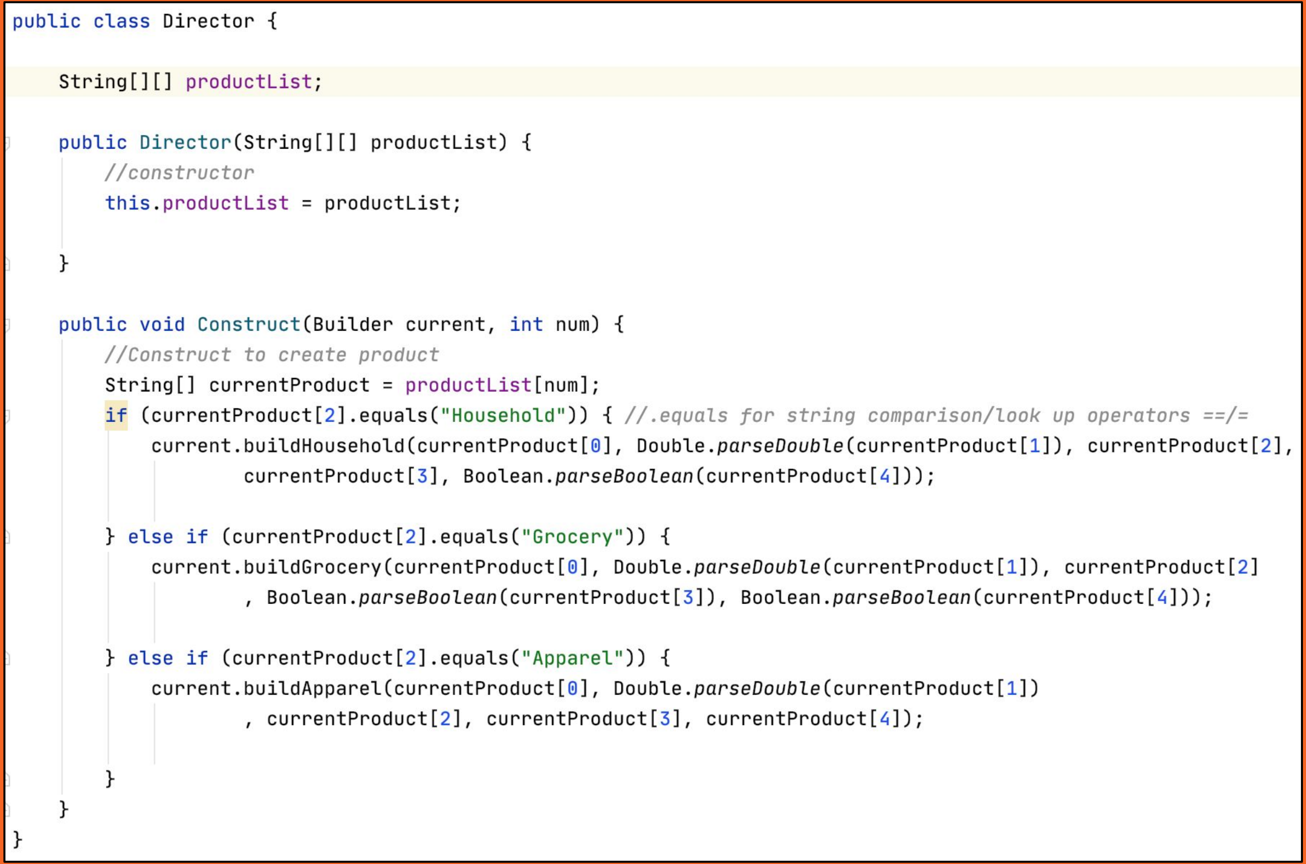
Builder Pattern

PROGRAM SNAPSHOT



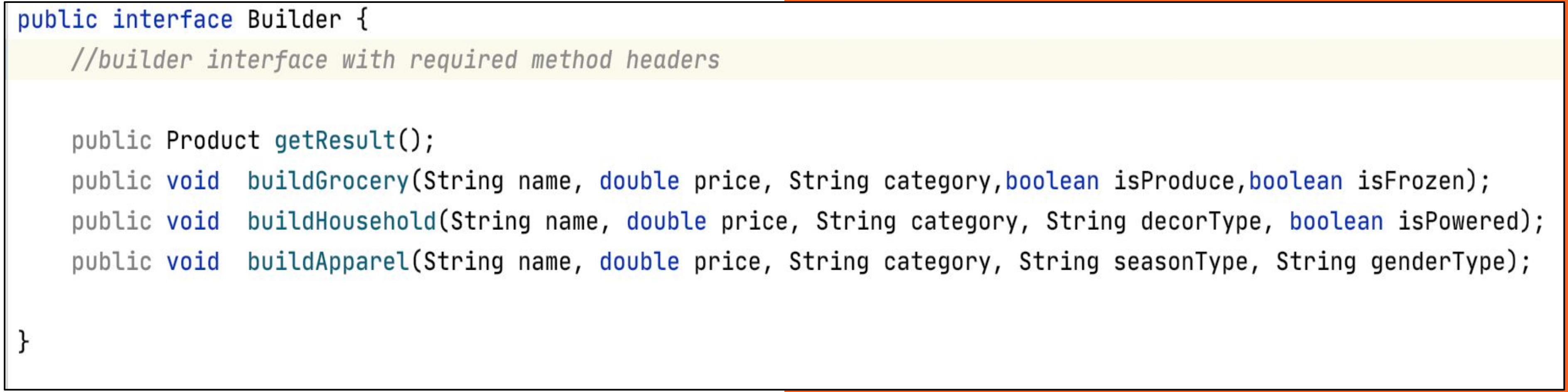
```
Welcome to Forum! Your one-stop shop for essentials. Please select your product from the following options to add to your cart:
0   Clorox Has been selected  Product Details:  Product: Clorox Price:  5.0 Category:  Household Decor Type: Cleaning Powered:  false
1   Windex Has been selected  Product Details:  Product: Windex Price:  5.0 Category:  Household Decor Type: Cleaning Powered:  false
2   Handbag Has been selected  Product Details:  Product: Handbag Price:  20.0 Category:  Apparel  Season Type: Spring Gender:  Female
Which item would you like to add to your cart? Please enter the corresponding number
 2
How many Handbag would you like to add?
3
3 Handbag has been added
What would you like to do now?
1: Add more products
2: Delete a product from your cart
3: Checkout
2
0   3x Handbag Has been selected  Product Details:  Product: Handbag Price:  20.0 Category:  Apparel  Season Type: Spring Gender:  Female
Which item would you like to remove from your cart? Please enter the corresponding number
1
How many would you like to remove?
1
What would you like to do now?
1: Add more products
2: Delete a product from your cart
3: Checkout
3
Your cart is ready for checkout
0   3x Handbag Has been selected  Product Details:  Product: Handbag Price:  20.0 Category:  Apparel  Season Type: Spring Gender:  Female
Your total is: $ 20.0
Thank you for shopping at Forum! We hope to see you again
```

Director

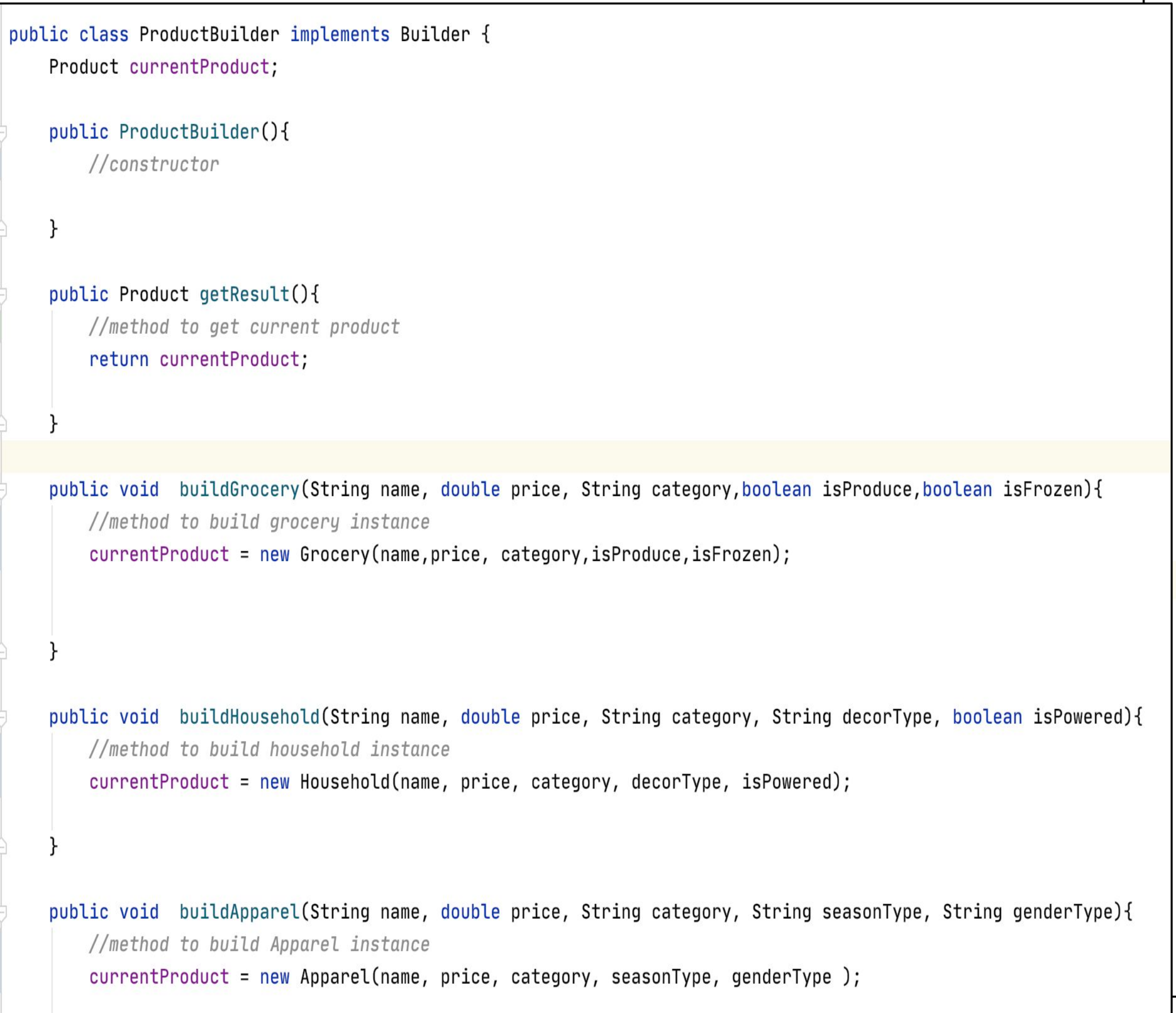


```java
public class Director {

    String[][] productList;

    public Director(String[][] productList) {
        //constructor
        this.productList = productList;

    }

    public void Construct(Builder current, int num) {
        //Construct to create product
        String[] currentProduct = productList[num];
        if (currentProduct[2].equals("Household")) { //.equals for string comparison/look up operators ==/=
            current.buildHousehold(currentProduct[0], Double.parseDouble(currentProduct[1]), currentProduct[2],
                    currentProduct[3], Boolean.parseBoolean(currentProduct[4]));

        } else if (currentProduct[2].equals("Grocery")) {
            current.buildGrocery(currentProduct[0], Double.parseDouble(currentProduct[1]), currentProduct[2]
                    , Boolean.parseBoolean(currentProduct[3]), Boolean.parseBoolean(currentProduct[4]));

        } else if (currentProduct[2].equals("Apparel")) {
            current.buildApparel(currentProduct[0], Double.parseDouble(currentProduct[1])
                    , currentProduct[2], currentProduct[3], currentProduct[4]);

        }
    }
}
```

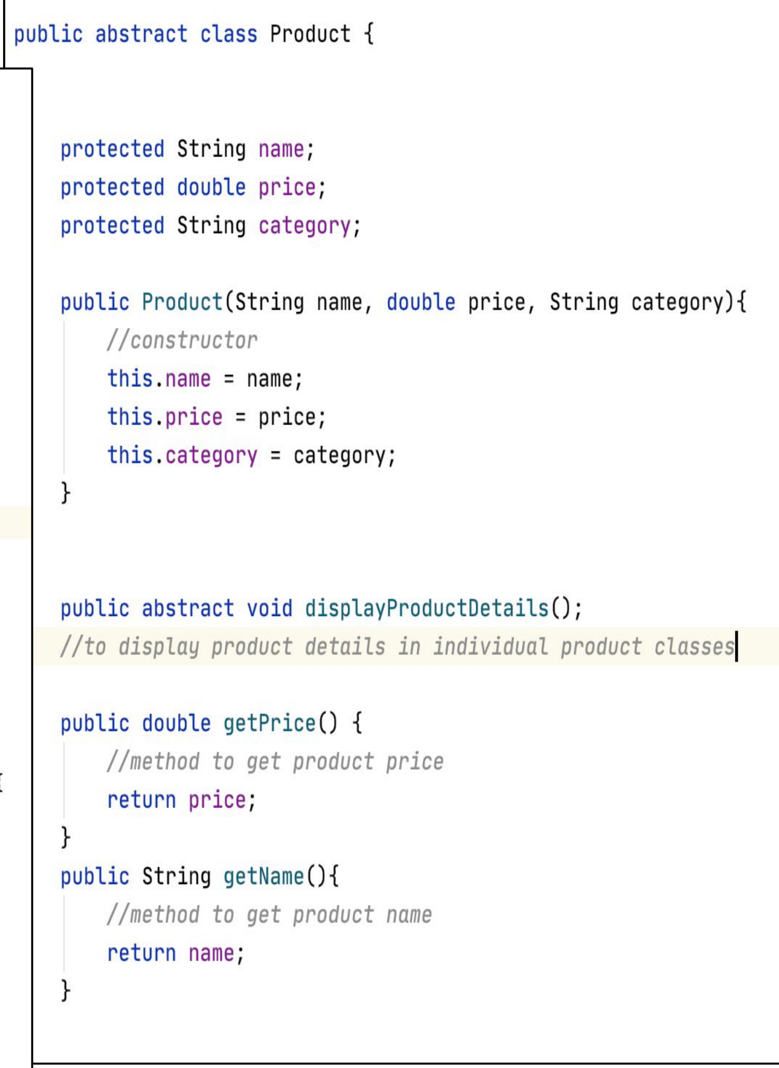
Builder Design Pattern - Components



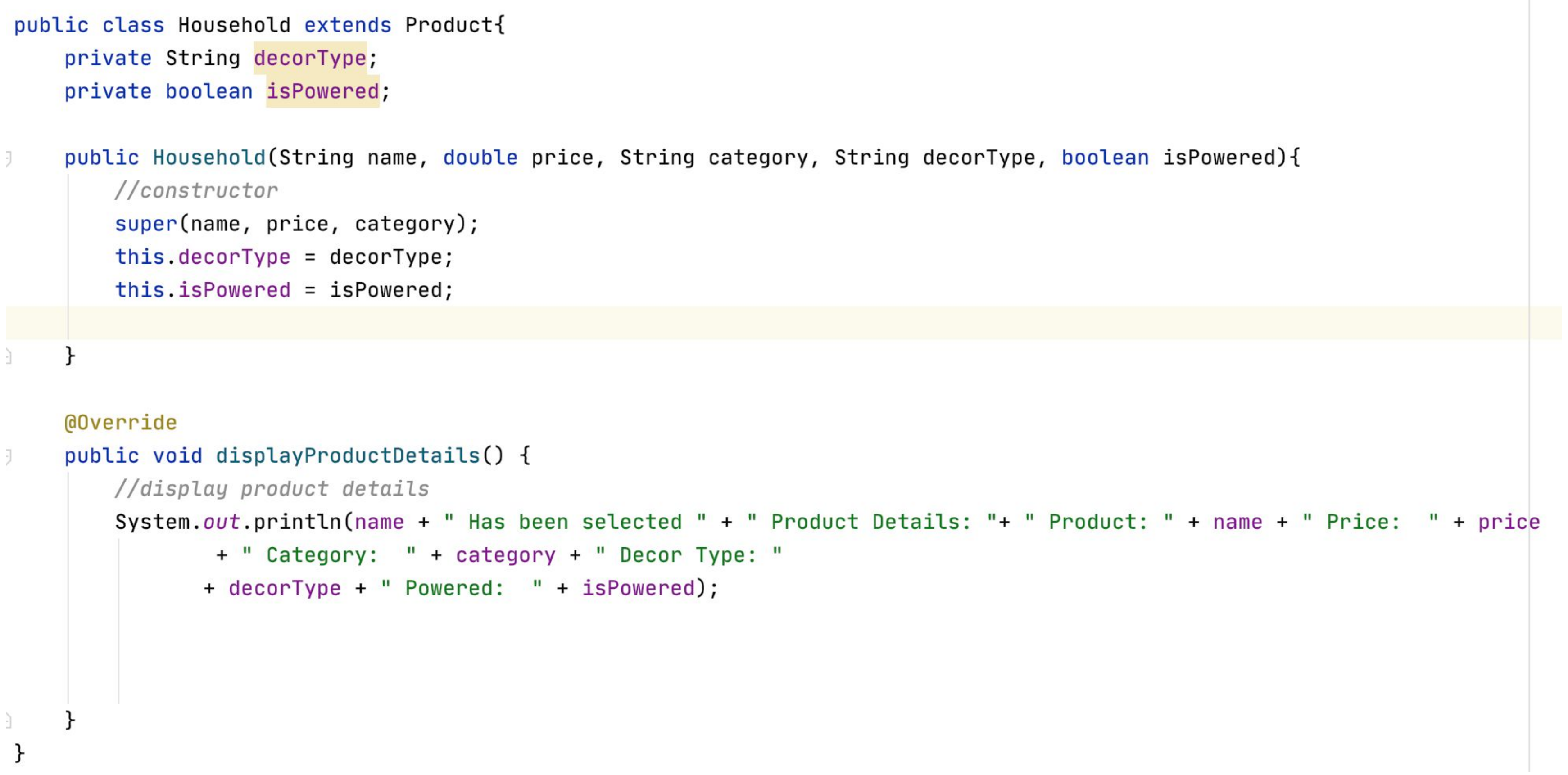
```java
public interface Builder {
    //builder interface with required method headers

    public Product getResult();
    public void  buildGrocery(String name, double price, String category,boolean isProduce,boolean isFrozen);
    public void  buildHousehold(String name, double price, String category, String decorType, boolean isPowered);
    public void  buildApparel(String name, double price, String category, String seasonType, String genderType);

}
```



```java
public class ProductBuilder implements Builder {
    Product currentProduct;

    public ProductBuilder(){
        //constructor


    }

    public Product getResult(){
        //method to get current product
        return currentProduct;

    }

    public void  buildGrocery(String name, double price, String category,boolean isProduce,boolean isFrozen){
        //method to build grocery instance
        currentProduct = new Grocery(name,price, category,isProduce,isFrozen);



    }

    public void  buildHousehold(String name, double price, String category, String decorType, boolean isPowered){
        //method to build household instance
        currentProduct = new Household(name, price, category, decorType, isPowered);


    }

    public void  buildApparel(String name, double price, String category, String seasonType, String genderType){
        //method to build Apparel instance
        currentProduct = new Apparel(name, price, category, seasonType, genderType );
```



```java
public abstract class Product {


    protected String name;
    protected double price;
    protected String category;

    public Product(String name, double price, String category){
        //constructor
        this.name = name;
        this.price = price;
        this.category = category;
    }



    public abstract void displayProductDetails();
    //to display product details in individual product classes

    public double getPrice() {
        //method to get product price
        return price;
    }
    public String getName(){
        //method to get product name
        return name;
    }
```



```java
public class Household extends Product{
    private String decorType;
    private boolean isPowered;

    public Household(String name, double price, String category, String decorType, boolean isPowered){
        //constructor
        super(name, price, category);
        this.decorType = decorType;
        this.isPowered = isPowered;

    }

    @Override
    public void displayProductDetails() {
        //display product details
        System.out.println(name + " Has been selected " + " Product Details: "+ " Product: " + name + " Price:  " + price
                + " Category:  " + category + " Decor Type: "
               + decorType + " Powered:  " + isPowered);



    }
}
```

Github Repo:
https://github.com/alina-akram/metcs-met-cs665-assignment-project-alina-akram

Thank you