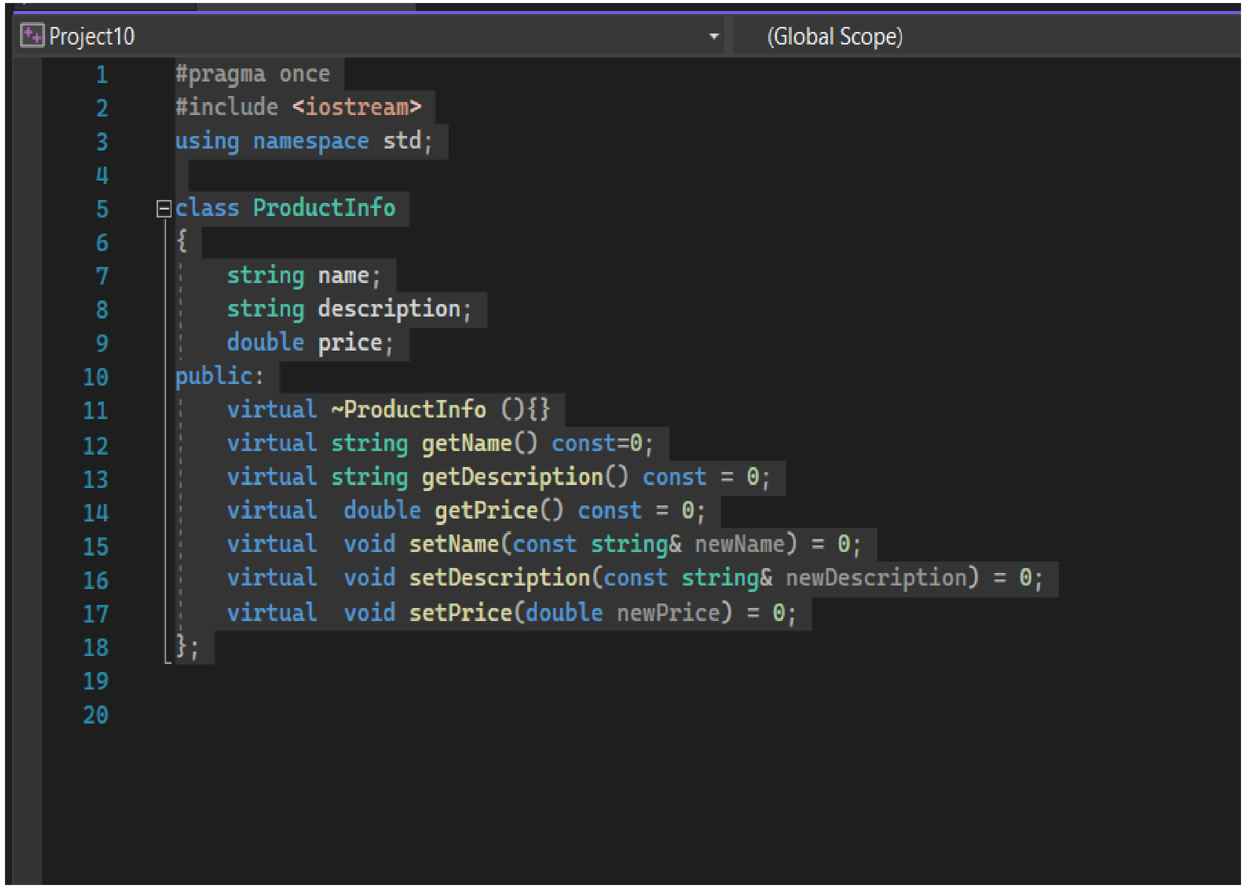
Design pattern project

- **Team members :**

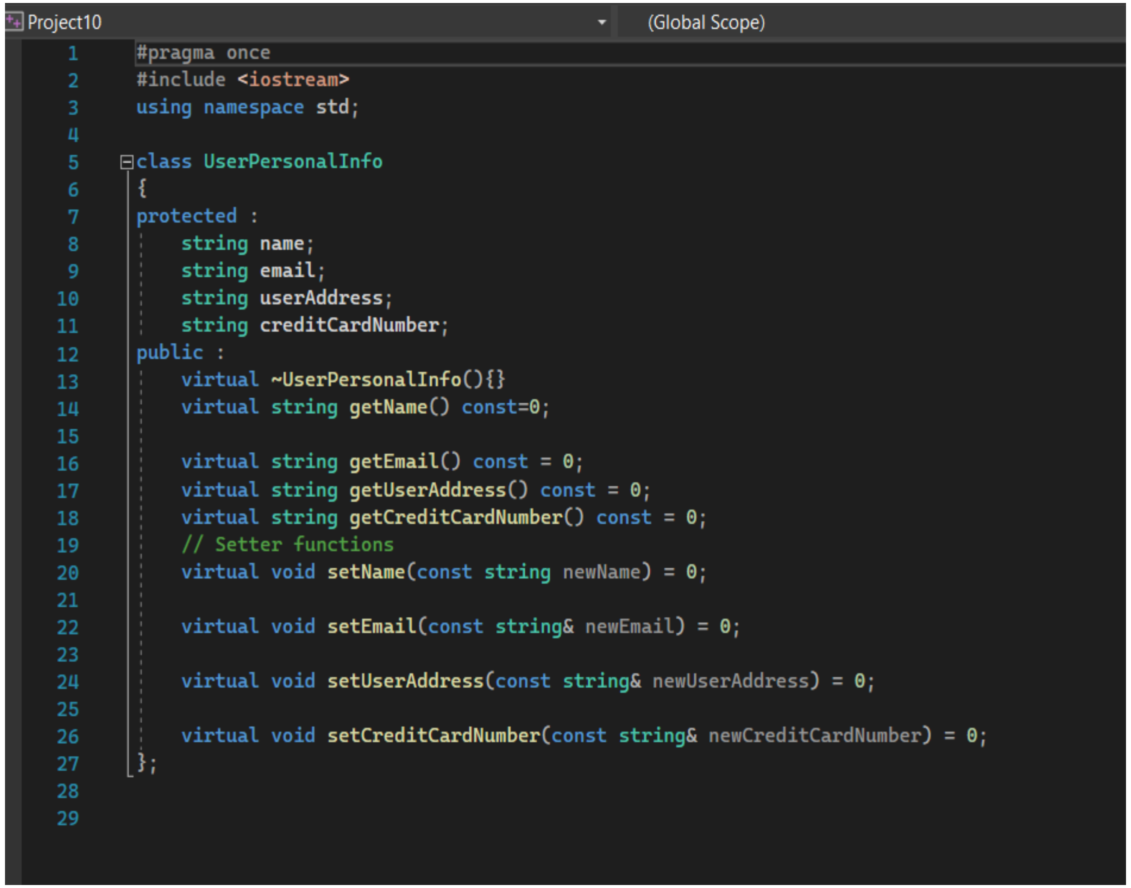
1-Asem Elenawy	21-101159
2- Abdullah Ayman	21-101131
3- Ahmed Tarek	21-101141
4- Ammar Wael	21-101152
5- Omar Essam	21-101029

Refactoring techniques :

Extract classes



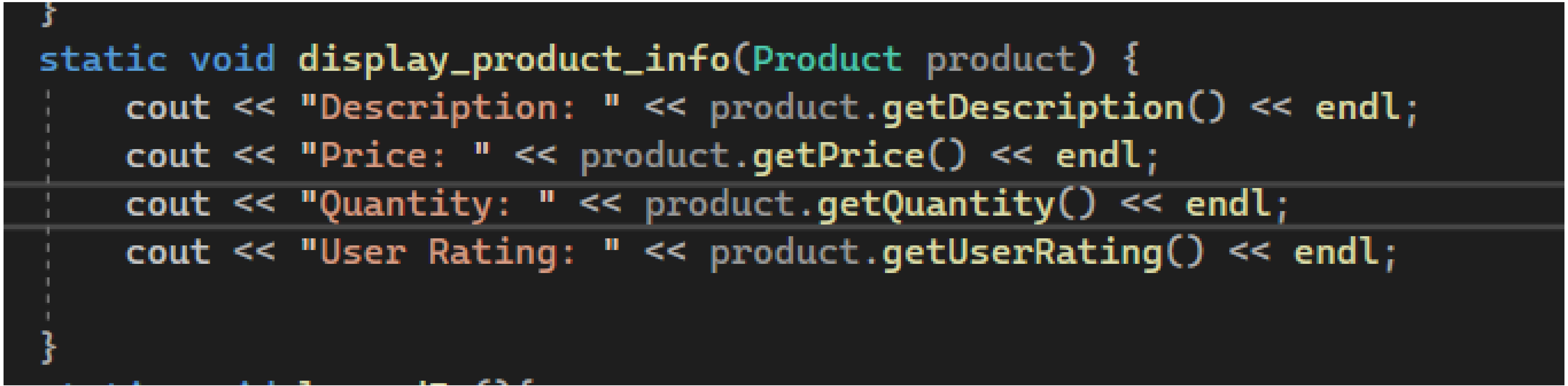
```cpp
#pragma once
#include <iostream>
using namespace std;

class ProductInfo
{
    string name;
    string description;
    double price;
public:
    virtual ~ProductInfo (){}
    virtual string getName() const=0;
    virtual string getDescription() const = 0;
    virtual  double getPrice() const = 0;
    virtual  void setName(const string& newName) = 0;
    virtual  void setDescription(const string& newDescription) = 0;
    virtual  void setPrice(double newPrice) = 0;
};
```



```cpp
#pragma once
#include <iostream>
using namespace std;

class UserPersonalInfo
{
protected :
    string name;
    string email;
    string userAddress;
    string creditCardNumber;
public :
    virtual ~UserPersonalInfo(){}
    virtual string getName() const=0;

    virtual string getEmail() const = 0;
    virtual string getUserAddress() const = 0;
    virtual string getCreditCardNumber() const = 0;
    // Setter functions
    virtual void setName(const string newName) = 0;

    virtual void setEmail(const string& newEmail) = 0;

    virtual void setUserAddress(const string& newUserAddress) = 0;

    virtual void setCreditCardNumber(const string& newCreditCardNumber) = 0;
};
```

Refactoring techniques :

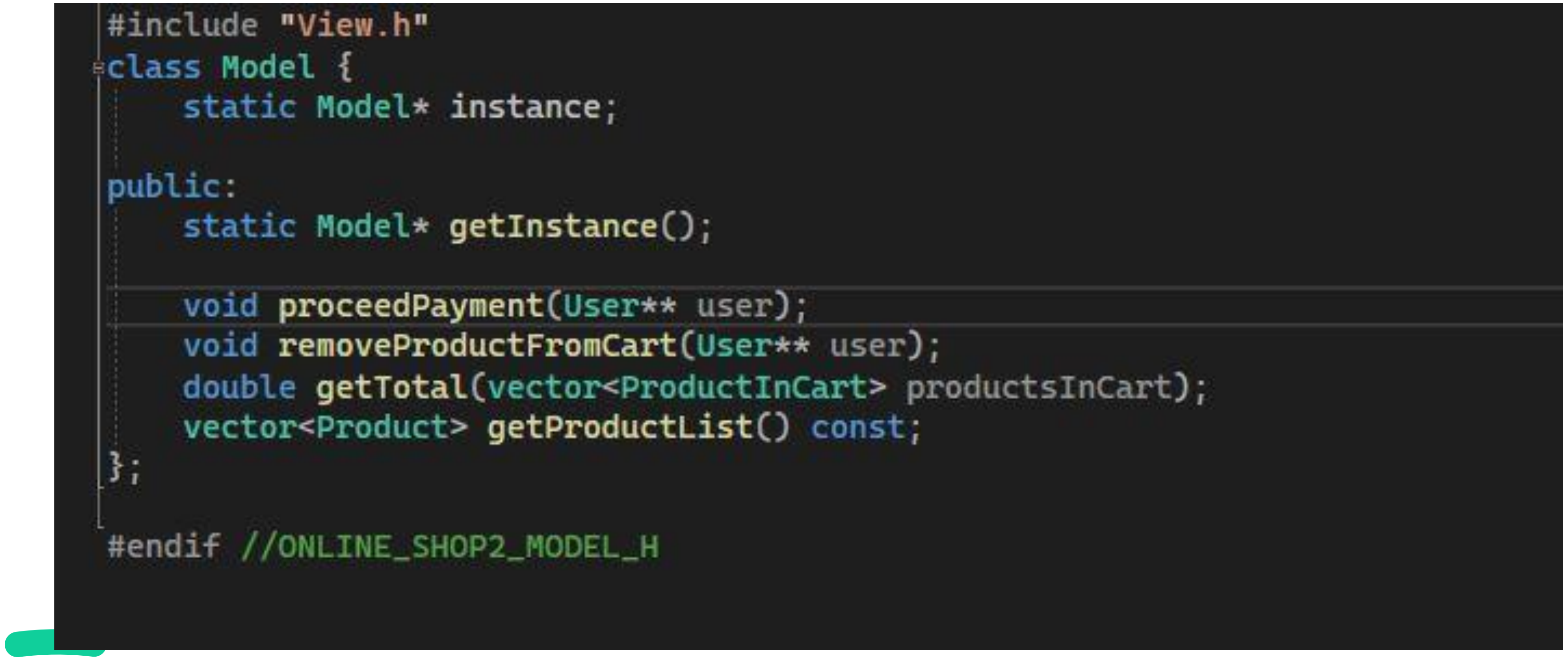
Extract methods



```
}
static void display_product_info(Product product) {
    cout << "Description: " << product.getDescription() << endl;
    cout << "Price: " << product.getPrice() << endl;
    cout << "Quantity: " << product.getQuantity() << endl;
    cout << "User Rating: " << product.getUserRating() << endl;

}
```

Design pattern ;

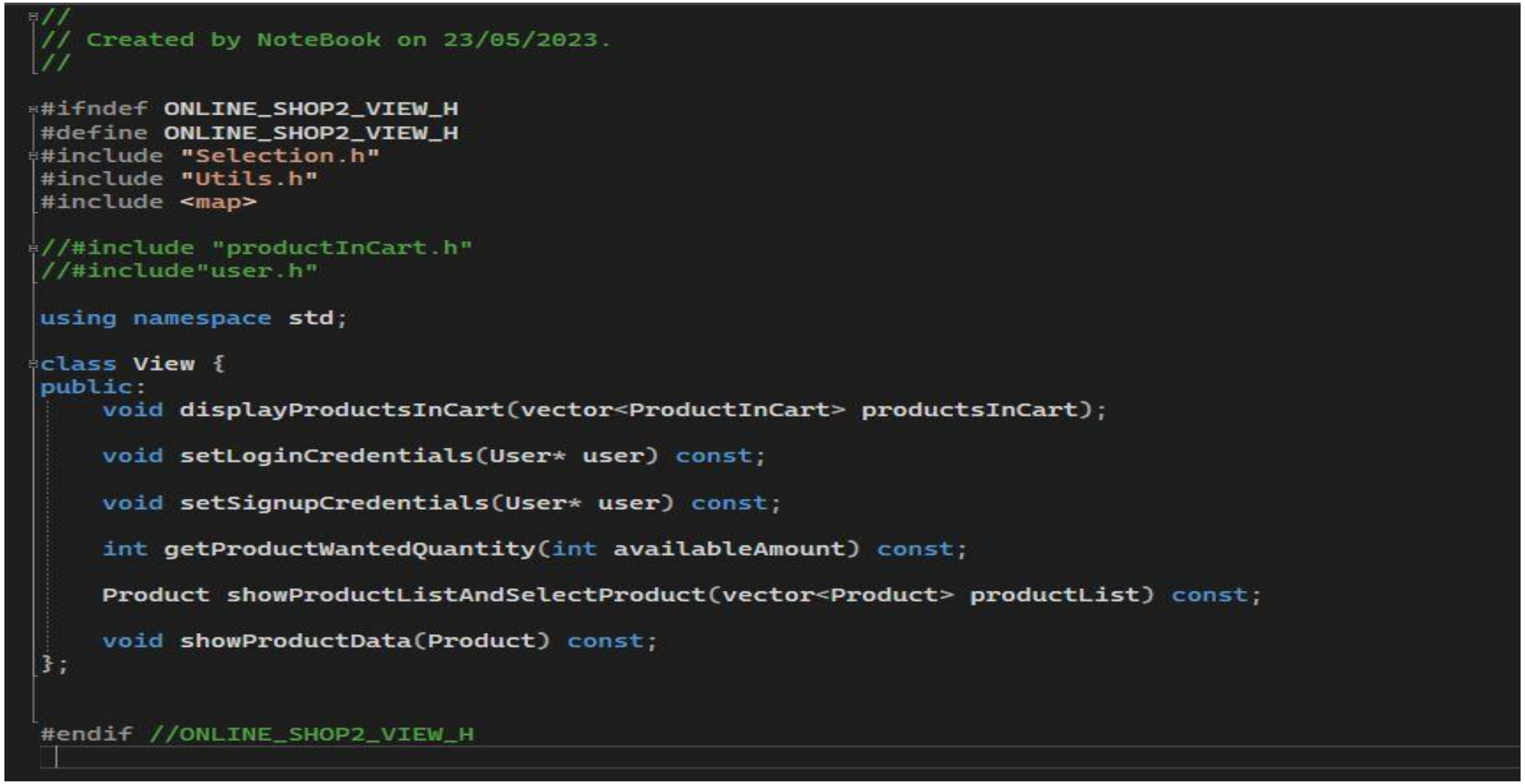
- **MVC -> Model**



```
#include "View.h"
class Model {
    static Model* instance;

public:
    static Model* getInstance();

    void proceedPayment(User** user);
    void removeProductFromCart(User** user);
    double getTotal(vector<ProductInCart> productsInCart);
    vector<Product> getProductList() const;
};

#endif //ONLINE_SHOP2_MODEL_H
```

Design pattern ;

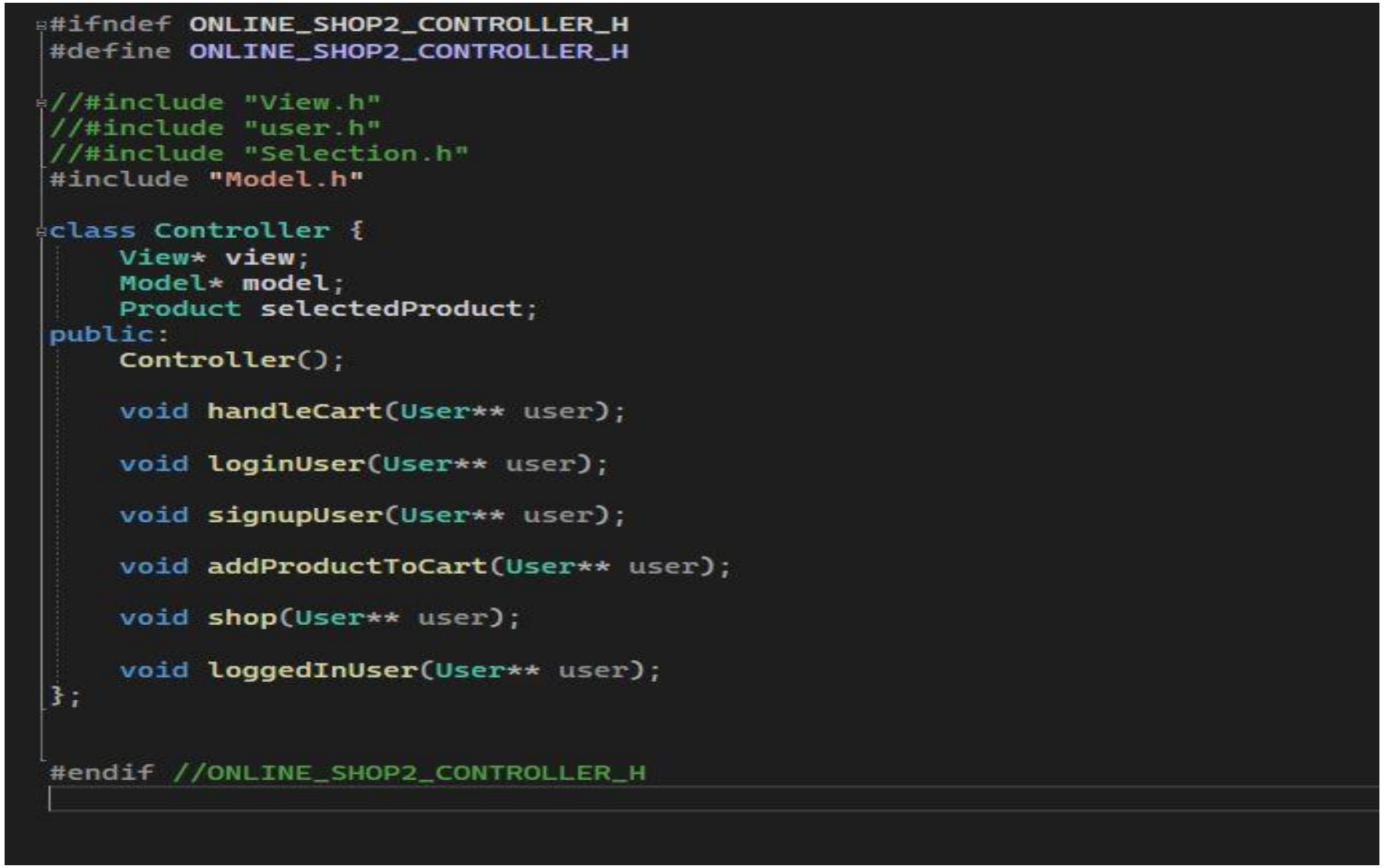
- **MVC -> view**



```
//
// Created by NoteBook on 23/05/2023.
//

#ifndef ONLINE_SHOP2_VIEW_H
#define ONLINE_SHOP2_VIEW_H
#include "Selection.h"
#include "Utils.h"
#include <map>

//#include "productInCart.h"
//#include"user.h"

using namespace std;

class View {
public:
    void displayProductsInCart(vector<ProductInCart> productsInCart);

    void setLoginCredentials(User* user) const;

    void setSignupCredentials(User* user) const;

    int getProductWantedQuantity(int availableAmount) const;

    Product showProductListAndSelectProduct(vector<Product> productList) const;

    void showProductData(Product) const;
};


#endif //ONLINE_SHOP2_VIEW_H
```

Design pattern ;

- **MVC -> controller**



```
#ifndef ONLINE_SHOP2_CONTROLLER_H
#define ONLINE_SHOP2_CONTROLLER_H

//#include "View.h"
//#include "user.h"
//#include "Selection.h"
#include "Model.h"

class Controller {
    View* view;
    Model* model;
    Product selectedProduct;
public:
    Controller();

    void handleCart(User** user);

    void loginUser(User** user);

    void signupUser(User** user);

    void addProductToCart(User** user);

    void shop(User** user);

    void loggedInUser(User** user);
};


#endif //ONLINE_SHOP2_CONTROLLER_H
```