**ASSIGNMENT 02**

OBJECT ORIENTED PROGRAMMING



GROUP MEMBERS:

|  |  |
| --- | --- |
| NAME | ENROLLMENT. NO |
| Saad Ahmad | 01-134222-130 |
| Syed Abdullah Shah | 01-134222-143 |
| Sohaib Ahmed | 01-134222-142 |

**DEPARTMENT OF COMPUTER SCIENCES**

**BAHRIA UNIVERSITY, ISLAMABAD CAMPUS**

***Question:*** Can you enhance this code by adding concept of Inheritance? If Yes, then which classes you will add and what functionalities you would like to add. (Write an Enhanced Complete code for this question). Consider writing all accessors and mutators along with other functions for all classes.

**Enhanced Code:**

//Syed Abdullah Shah

//Saad Ahmed

//Sohaib Ahmed

#include<iostream>

#include<string>

using namespace std;

class Date {

protected:

int day;

int month;

int year;

public:

//Default Value Constructor

Date(int tempDate=1, int tempMonth=1, int tempYear=1) : day(tempDate), month(tempMonth), year(tempYear) {}

//Setter Functions

void set\_day(int day)

{

this->day = day;

}

void set\_month(int month)

{

this->month = month;

}

void set\_year(int year)

{

this->year = year;

}

//Getter Functions

int get\_day() const

{

return day;

}

int get\_month() const

{

return month;

}

int get\_year() const

{

return year;

}

};

class Details {

protected:

string name;

int ID;

public:

//Default value constructor

Details(string tempName="", int tempID=1) :name(tempName), ID(tempID) {}

//Setter Functions

void set\_name(string name)

{

this->name = name;

}

void set\_id(int id)

{

ID = id;

}

//Getter Functions

string get\_name() const

{

return name;

}

int get\_id() const

{

return ID;

}

};

class Customer : public Details

{

private:

string email;

string address;

public:

//Default Value Constructor

Customer(int tempID=0, string tempName=" ", string tempAddress=" ", string tempEmail=" ") : Details(tempName, tempID)

{

set\_email(tempEmail);

validateEmail();

set\_address(tempAddress);

}

//Setter Functions

void set\_email(string email)

{

this->email = email;

}

void set\_address(string address)

{

this->address = address;

}

//Getter Functions

string get\_email() const

{

return email;

}

string get\_address() const

{

return address;

}

void validateEmail() { //Function to check if the email has @ and . in it

int i, j;

bool check;

do {

int length = email.length();

check = false;

for (int i = 0; i < length; i++)

{

if (email[i] == '@')

{

for (int j = 0; j < length; j++) {

if (email[j] == '.')

{

check = true;

break;

}

}

}

}

if (check == false)

{

cout << "Enter a valid email address with \"@\" and \".\" " << endl;

cin >> email;

}

} while (check != true);

}

};

class Item : public Details {

protected:

double itemPrice;

public:

Item(int tempID=1, string tempName=" ", double tempPrice=0) : Details(tempName, tempID) {

set\_item\_price(tempPrice);

}

//Setter Function

void set\_item\_price(double item\_price)

{

itemPrice = item\_price;

}

//Getter Function

double get\_item\_price() const

{

return itemPrice;

}

};

class Order : public Item {

private:

int orderID;

Customer customer;

Date orderDate;

double orderTotal;

int quantity;

public:

//Empty Constructor

Order();

//Parameterized Constructor

Order(int tempID, Customer tempCustomer, Date tempOrderDate, Item tempItem)

: orderID(tempID),

customer(tempCustomer),

orderDate(tempOrderDate),

Item(tempItem)

{

}

//Member Functions

void addOrderDetail(Item tempItem, int tempQuantity)

{

name = tempItem.get\_name();

itemPrice = tempItem.get\_item\_price();

set\_quantity(tempQuantity);

}

void calculateTotal()

{

set\_order\_total(get\_item\_price() \* quantity);

}

void displayOrderDetails() {

cout << " Order Details" << endl << endl;

cout << " ------------------------------------------------" << endl << endl;

cout << " Name : " << customer.get\_name() << endl << endl;

cout << " Email : " << customer.get\_email() << endl << endl;

cout << " Address : " << customer.get\_address() << endl << endl;

cout << " Item Name : " << name << endl << endl;

cout << " Item Price : " << itemPrice << endl << endl;

cout << " Item Quantity : " << get\_quantity() << endl << endl;

cout << " Total Bill : " << get\_order\_total() << endl << endl;

cout << " Date : " << orderDate.get\_day() << "/" << orderDate.get\_month() << "/" << orderDate.get\_year() << endl << endl;

cout << " ------------------------------------------------" << endl << endl;

}

//Setter Functions

void set\_quantity(int quantity)

{

this->quantity = quantity;

}

void set\_order\_id(int order\_id)

{

orderID = order\_id;

}

void set\_order\_total(double order\_total)

{

orderTotal = order\_total;

}

//Getter Functions

int get\_quantity() const

{

return quantity;

}

int get\_order\_id() const

{

return orderID;

}

double get\_order\_total() const

{

return orderTotal;

}

};

class Payment {

private:

int paymentID;

Order order;

double paymentAmount;

Date payment\_Date;

public:

//Empty Constructor

Payment();

//Parameterized Constructor

Payment(int payment\_id, Order order, double payment\_amount, Date pDate)

:paymentID(payment\_id),

order(order),

paymentAmount(payment\_amount),

payment\_Date(pDate)

{

}

//Member Functions

void p\_Date() const

{

cout << payment\_Date.get\_day() << "/" << payment\_Date.get\_month() << "/" << payment\_Date.get\_year() << endl;

}

//Setter Functions

void set\_payment\_id(int payment\_id)

{

paymentID = payment\_id;

}

void set\_payment\_amount(double payment\_amount)

{

paymentAmount = payment\_amount;

}

//Getter Functions

int get\_payment\_id() const

{

return paymentID;

}

double get\_payment\_amount() const

{

return paymentAmount;

}

};

int main() {

Customer customer(1, "Saad Ahmad", "House#123,Street#123", "saad9009@gmail.com"); //if @ and . are not used in email address, then ask again for customer email.

Item item(1, "PS5", 150000);

Date OrderDate(8, 5, 2023);

Date PaymentDate(13, 5, 2023);

Order order(1, customer, OrderDate, item);

order.addOrderDetail(item, 2); // 2 is quantity

order.calculateTotal();

order.displayOrderDetails();

Payment payment(1, order, 300000, PaymentDate);

cout << " Payment Details" << endl << endl;

cout << " ------------------------------------------------" << endl << endl;

cout << " Payment ID : " << payment.get\_payment\_id() << endl << endl;

cout << " Payment Amount : " << payment.get\_payment\_amount() << endl << endl;

cout << " Payment Date : ";

payment.p\_Date();

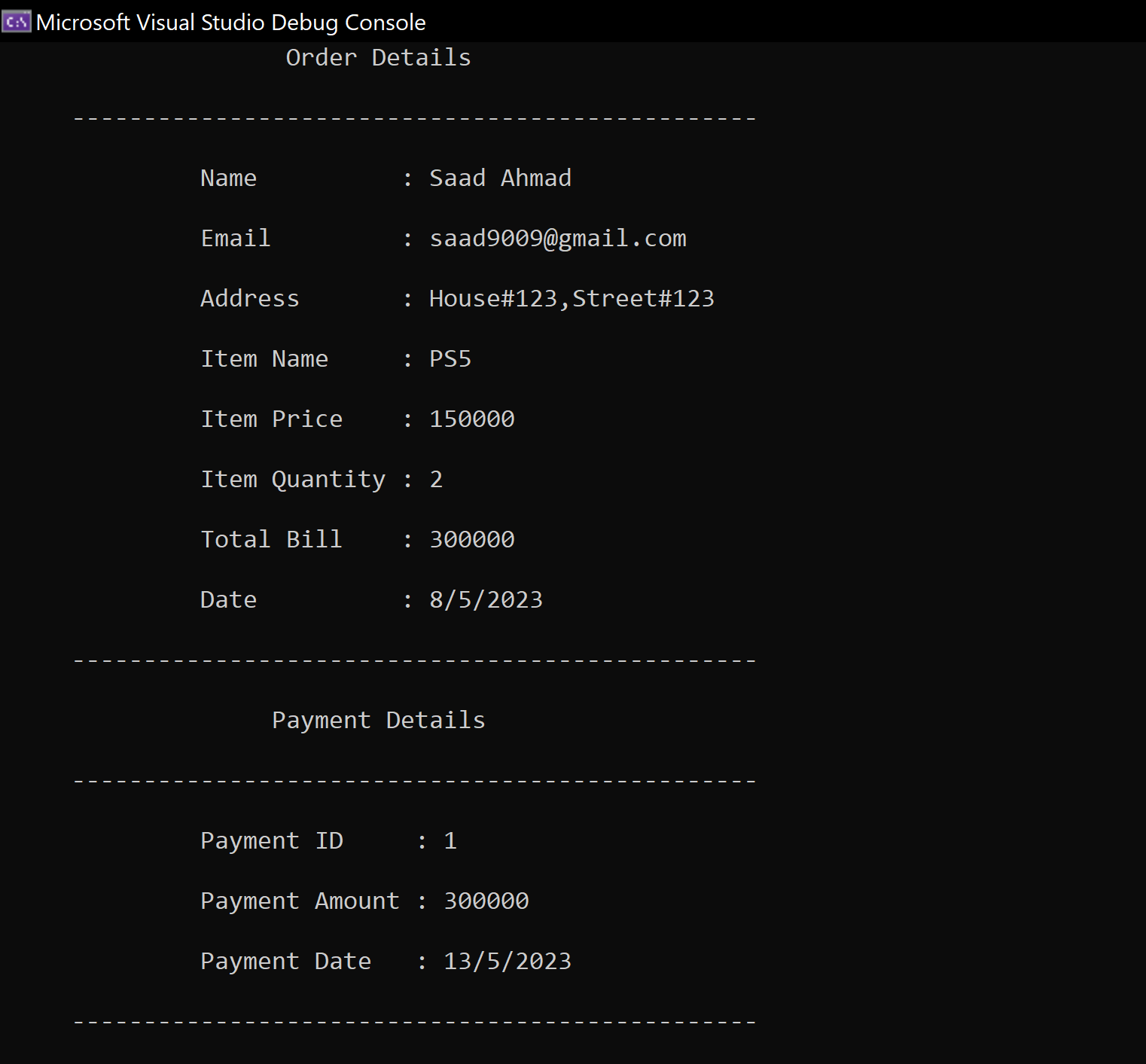
cout << endl;

cout << " ------------------------------------------------" << endl << endl;

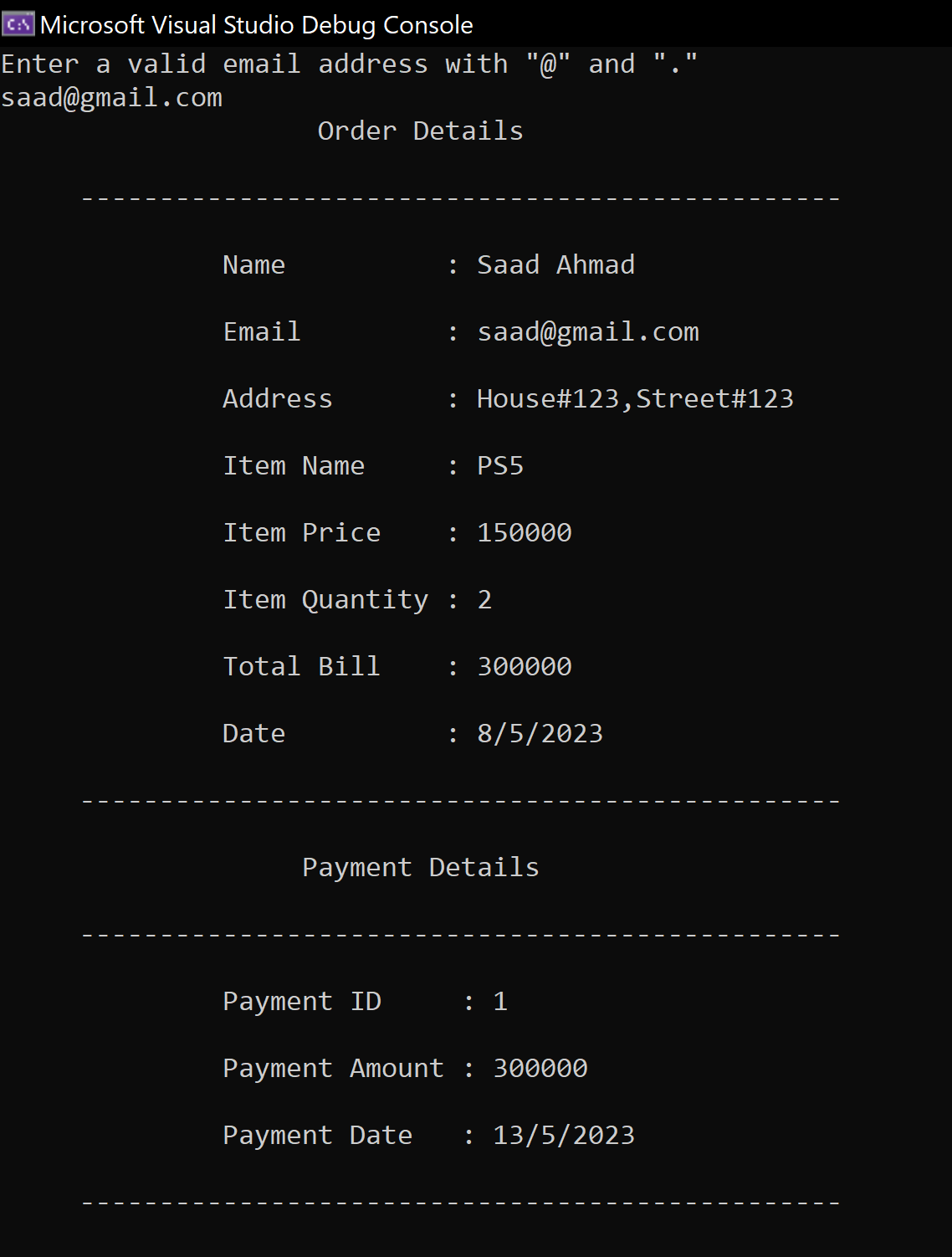
return 0;

}

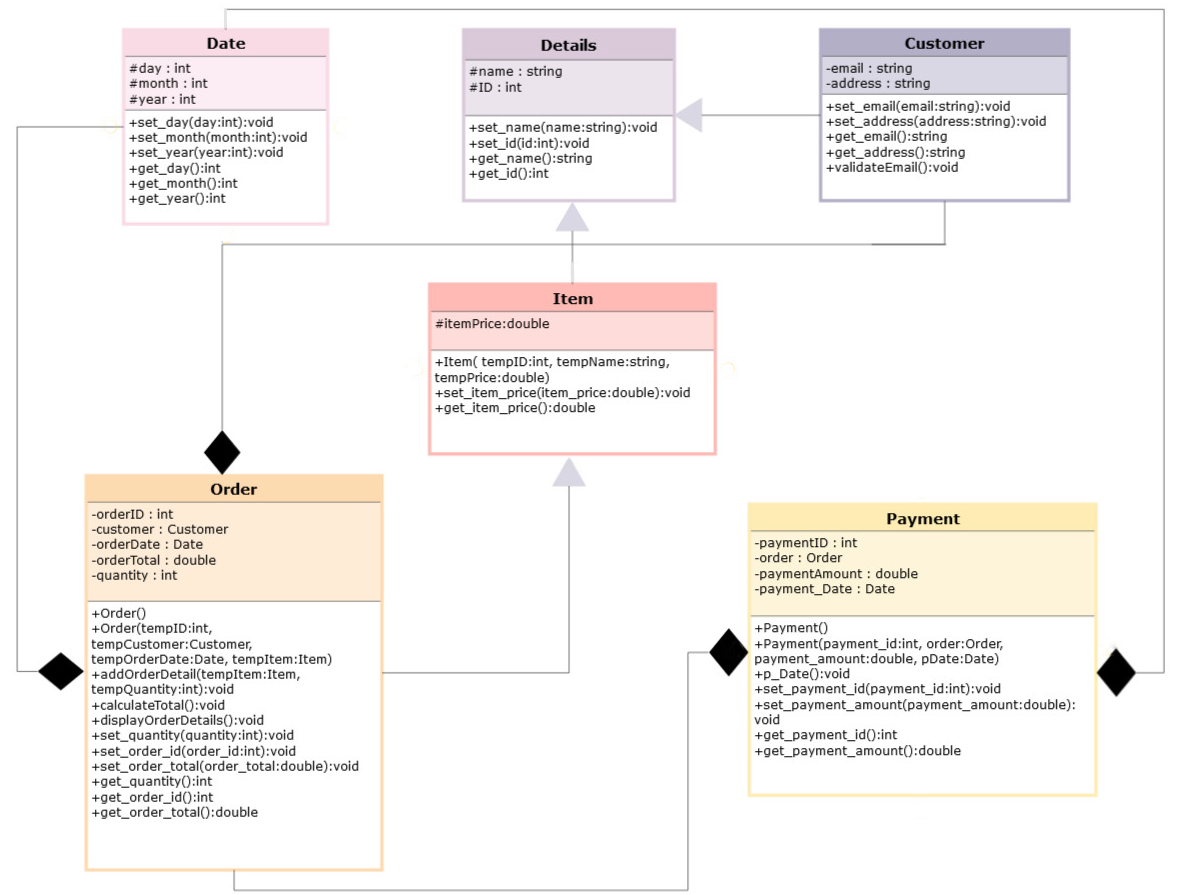
Screen Shots:

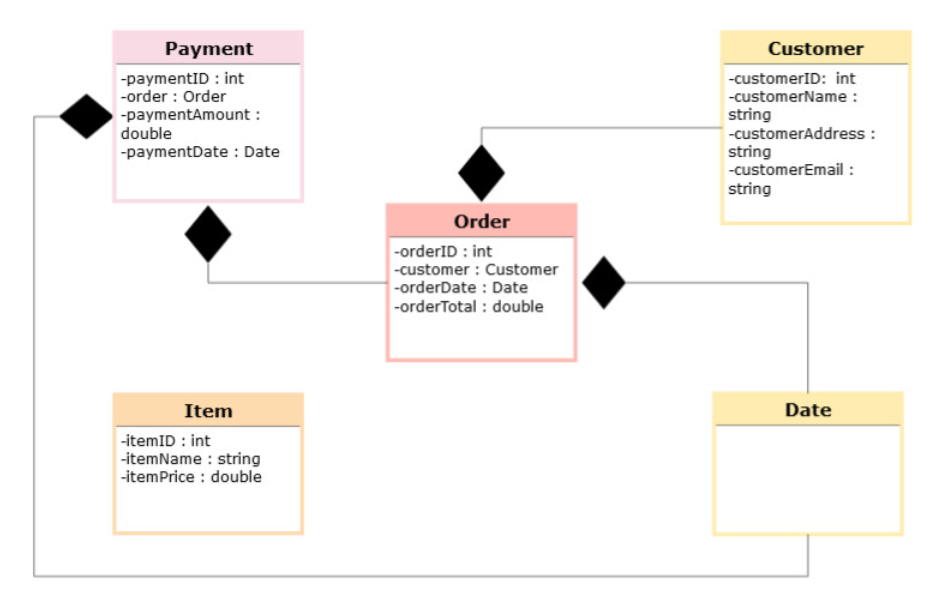


Invalid Email:



**UML Diagram:**

****

****