OOP3

Sales Order App

—

Your Name: **Saeed Ahmed Khadra**
Your ID: 20912021100427
Your Section: 12

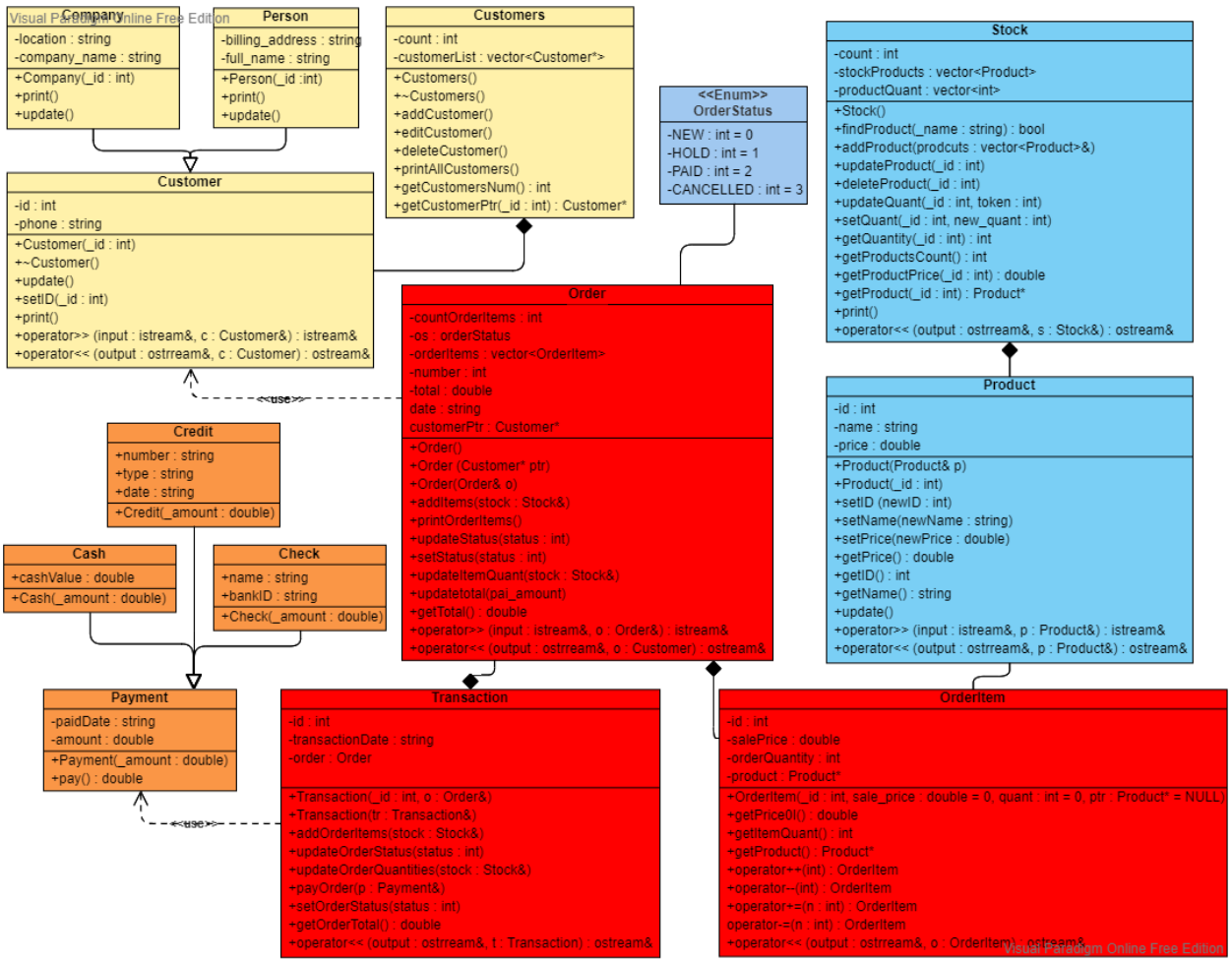
Overview

Application Description:

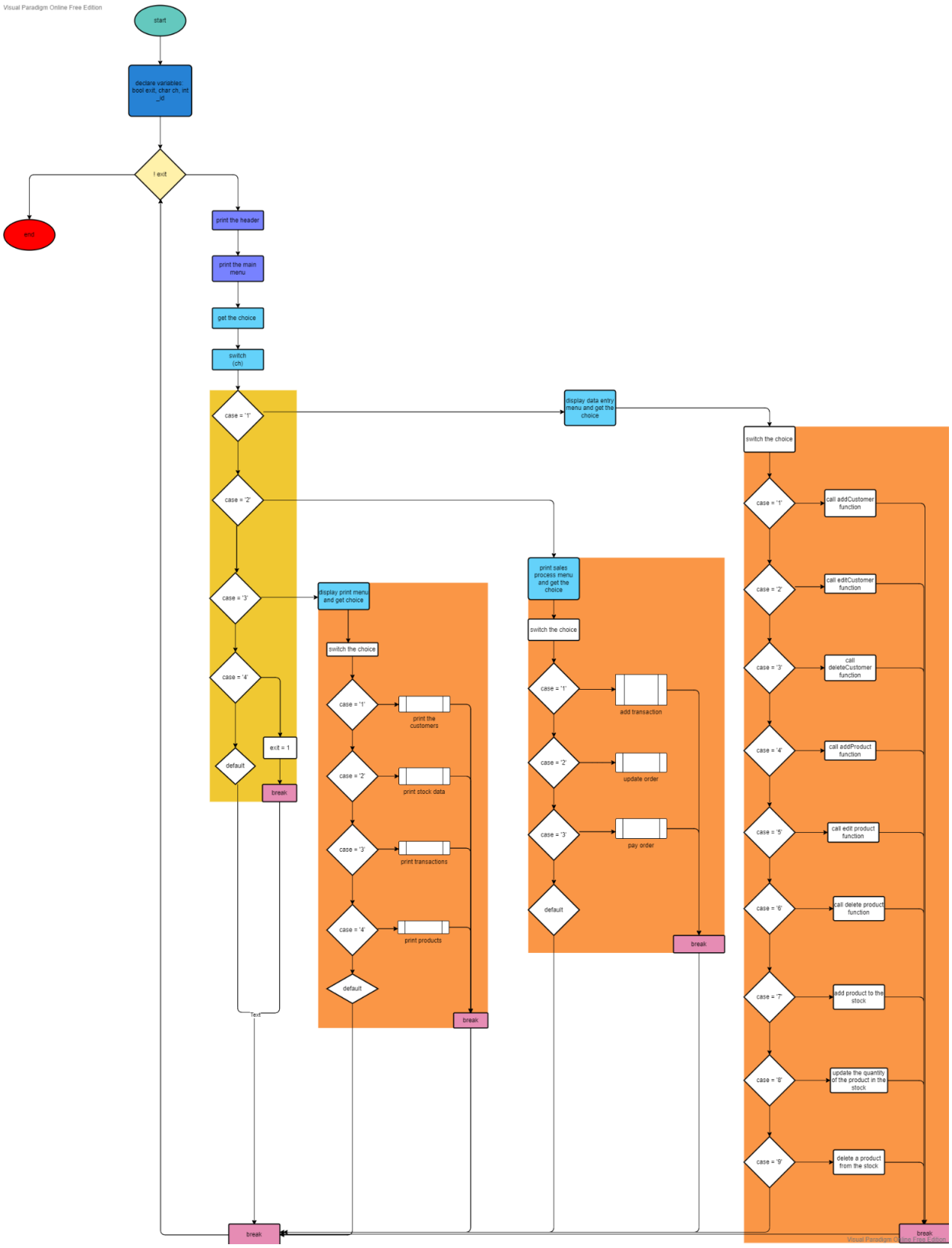
This app allows you to create a stock and store products in it,
Deal with the customers for both persons and companies,
Make transactions for the orders (which contain multiple items with the ability to add and edit the quantities of them)
And finally paying the transaction with one of those methods: credit, cash and check.

Diagrams

1. Class Diagram



2. Flow Diagram



Data Structure (Data View)

Stock&Product:

- There is a Product List in the main, and a Stock Product List in the Stock that contain a sublist of the main list.

Customers:

- There is a Customer List in “Customers” class, which contain customers of both types: Person and Company.

Transactions:

- There is a Transaction List in the main, each element contain one Order, which contain an Order Item List.

Classes & Functions

1. **Stock**

	Function Name	Description	Input	Output
1	Find Product	Check if the given product name exists in the product list or not.	Product name	“True” if found, “false” if not.
2	Add Product	Select a product from the main product list to add to the stock.	Main product list (by reference).	void
3	Update product	Updates product name and price.	The id of the product.	void
4	Delete product	Deletes a product from the stock.	The id of the product.	void

5	Update quant	Updates the quantity of the product in the stock.	The id of the product + the token quantity (which was selled).	void
6	Print	Prints the stock's attributes.	void	The stock's data.
7	Operator <<	Prints the stock's attributes.	Reference to Ostream + reference to Stock.	The stock's data.

2. **Product**

	Function Name	**Description**	**Input**	**Output**
1	update	Updates the product quantity and name.	void	void
2	Operator >>	Get the product infos from the user.	Reference to Istream + reference to Product.	void
3	Operator <<	Prints the product's attributes.	Reference to Ostream + reference to product.	Product's infos.

3. **Person**

	Fucntion Name	**Description**	**Input**	**Output**
1	print	Prints the person's attributes.	void	Person's attributes.
2	update	Updates the person's attributes.	void	void

4. **Company**

	Fucntion Name	Description	Input	Output
1	print	Prints the company's attributes.	void	company's attributes.
2	update	Updates the company's attributes.	void	void

5. Customer

	Function Name	Description	Input	Output
1	~Customer	Virtual destrtuctor to free the object allocated using "new"	void	void
2	update	Updates the customer's infos.	void	void
3	Operator >>	Get the customer's infos from the user.	Reference to Istream + reference to Customer.	Customer's infos.
4	Operator <<	Prints the customer's attributes.	Reference to Ostream + reference to customer.	void

6. Customers

	Function Name	Description	Input	Output
1	~Customers	Destructor to free up the pointers in customer list.	void	void
2	Add Customer	Add a new customer to the customers list.	void	void
3	Edit Customer	Update the infos of a specific customer.	void	void
4	Delete Customer	Delete a specific	void	void

		customer from the list.		
5	Print all customers	Prints all customers in my customers list.	void	Customers infos.

7. **Order Item**

	Function Name	Description	Input	Output
1	Operator ++	Increase the quantity of the item by 1	Dummy para (int)	void
2	Operator - -	Decrease the quantity of the item by 1	Dummy para (int)	void
3	Operator +=	Increase the quantity by n	Int n	void
4	Operator -=	Decrease the quantity by n	Int n	void
5	Operator <<	Prints the item's infos.	Ostream reference + Order Item reference.	Order item infos.

8. **Order**

	Function Name	Description	Input	Output
1	Add Items	Add items to my order item list.	Reference to the Stock.	void
2	Print items	Prints the infos of all order item list elements.	void	Order items infos.
3	Update status	Updates the Order's status.	Int status	void
4	Update Item quant	Updates the quantity of the order item (depending on the quantity of this item in the Stock)	Reference to Stock.	void

5	Update total	Updates the total amount of the order.	Double : paid_amount	void
6	Operator >>	Get the Order's infos from the user.	Reference to Istream + reference to Order.	Order's infos.
7	Operator <<	Prints the Order's attributes.	Reference to Ostream + reference to Order.	void

9. Transaction

	Function Name	**Description**	**Input**	**Output**
	Add order items	Used to add order items to the order of the transaction	Stock reference	void
	Update order status	To update the status of the order.	Int status	void
	Update order quantities	Update the quantity of an order item of the order.	Stock reference	void
	Pay order	To pay the order using any of the three methods	Payment reference	void
	Get order total	To get the remaining amount that didn't paid yet.	void	Return the remaining total of the order.
	Operator <<	Prints the attributes of the transaction.	Ostream reference + Transaction reference.	The current transaction's infos.

10. Cash

	Function Name	Description	Input	Output
	Cash	constructor to Set the cash value	Double _amount	void

11. Check

	Function Name	Description	Input	Output
	Check	constructor to set the amount of the payment	Double _amount	void

12. Credit

	Function Name	Description	Input	Output
	Credit	Construtor to set the amount of the payment.	Double _amount	void

13. Payment

	Function Name	Description	Input	Output
	Payment	constructor to set the amount	Double _amount	void
	pay	To pay the amount for an order	void	Return the amount value.

Demo

1. **Scienaro Case:**

Input 1 to pen the data entry menu,

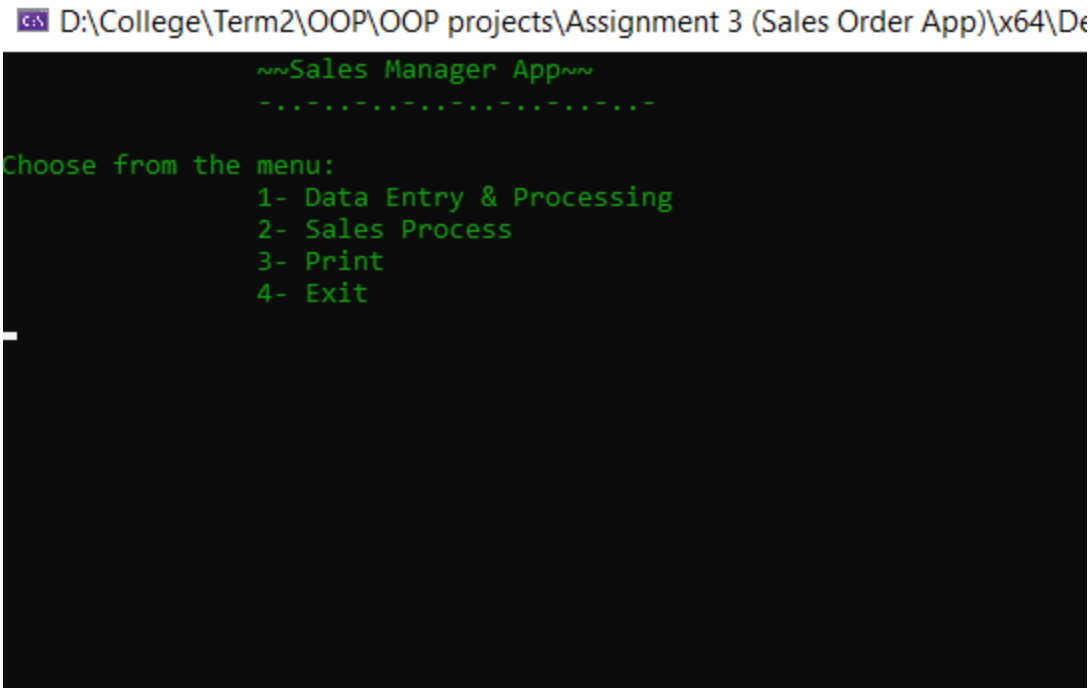
Then input 4 to add an abstract product,

Then enter the product infos,

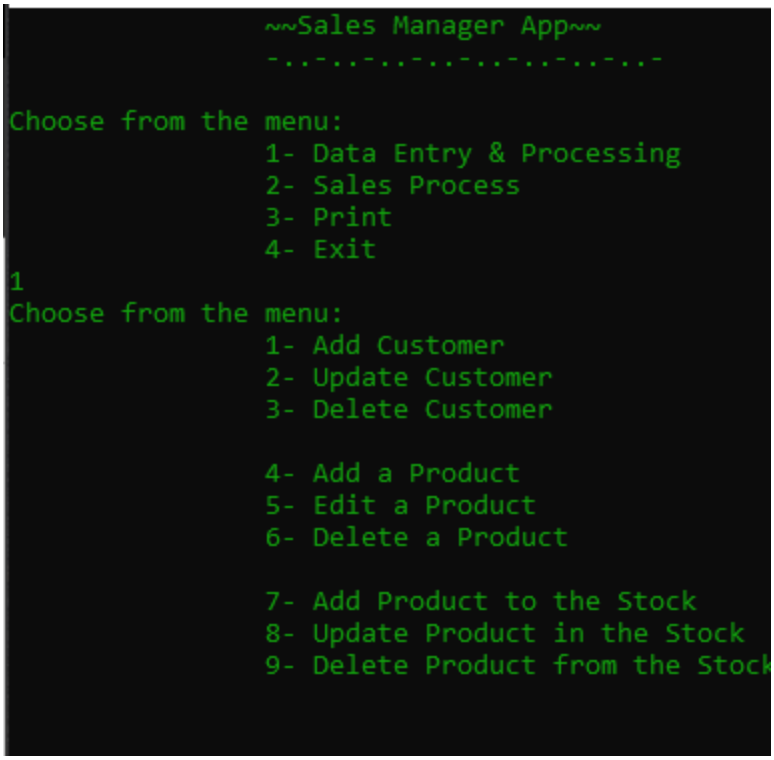
Then press enter to save.

2. Screenshots:

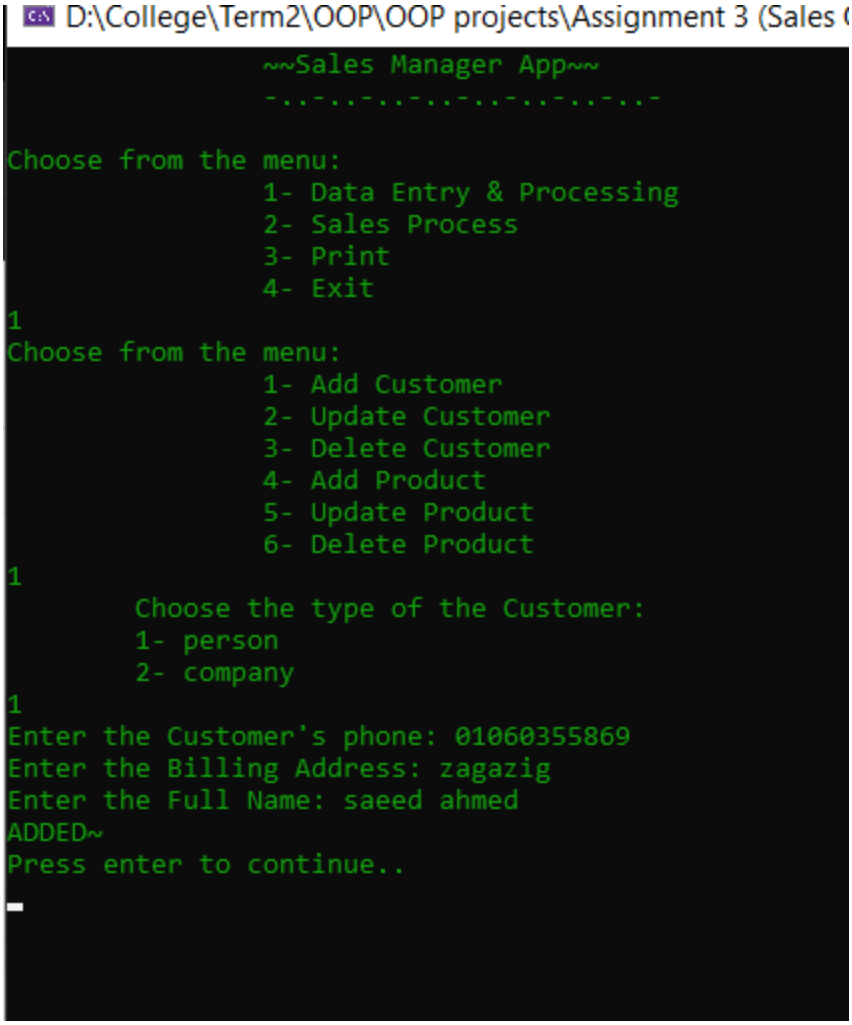
- Main menu:



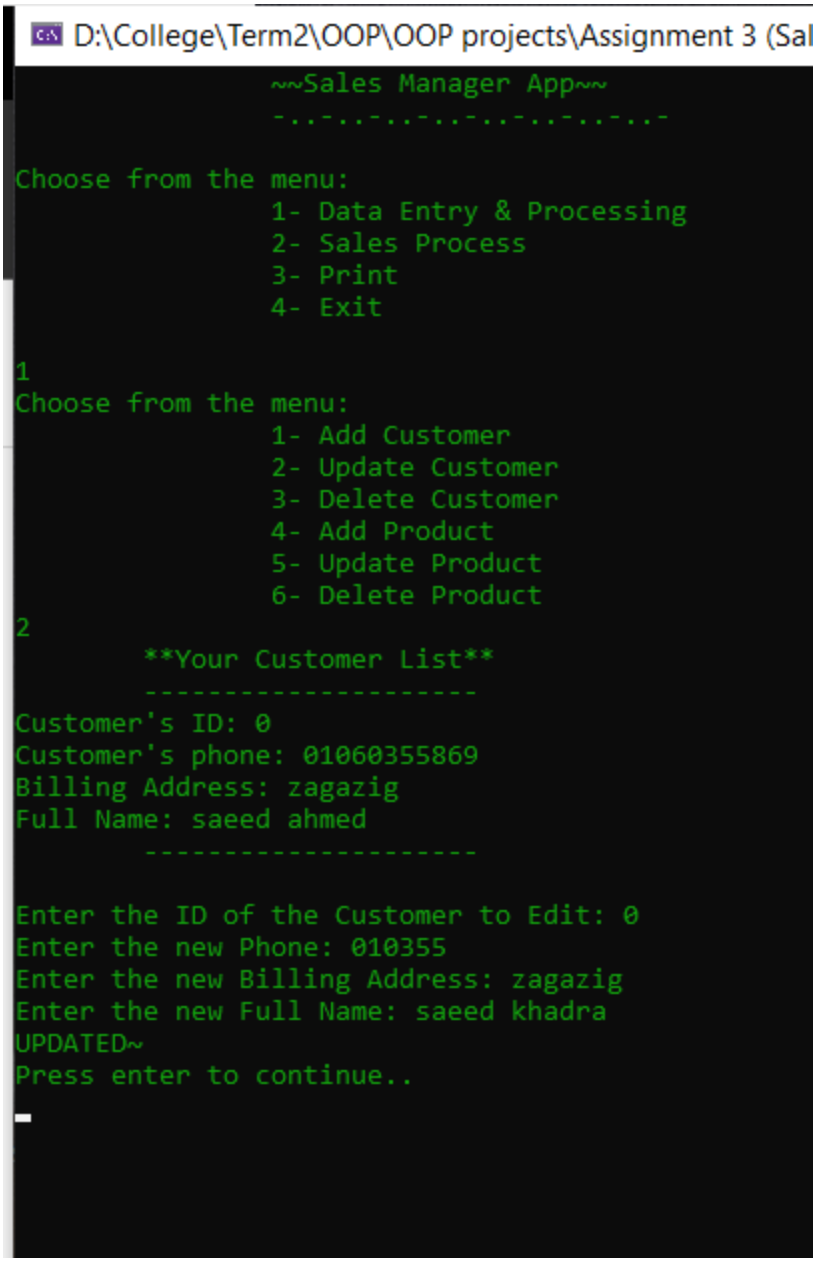
- Data entry menu:



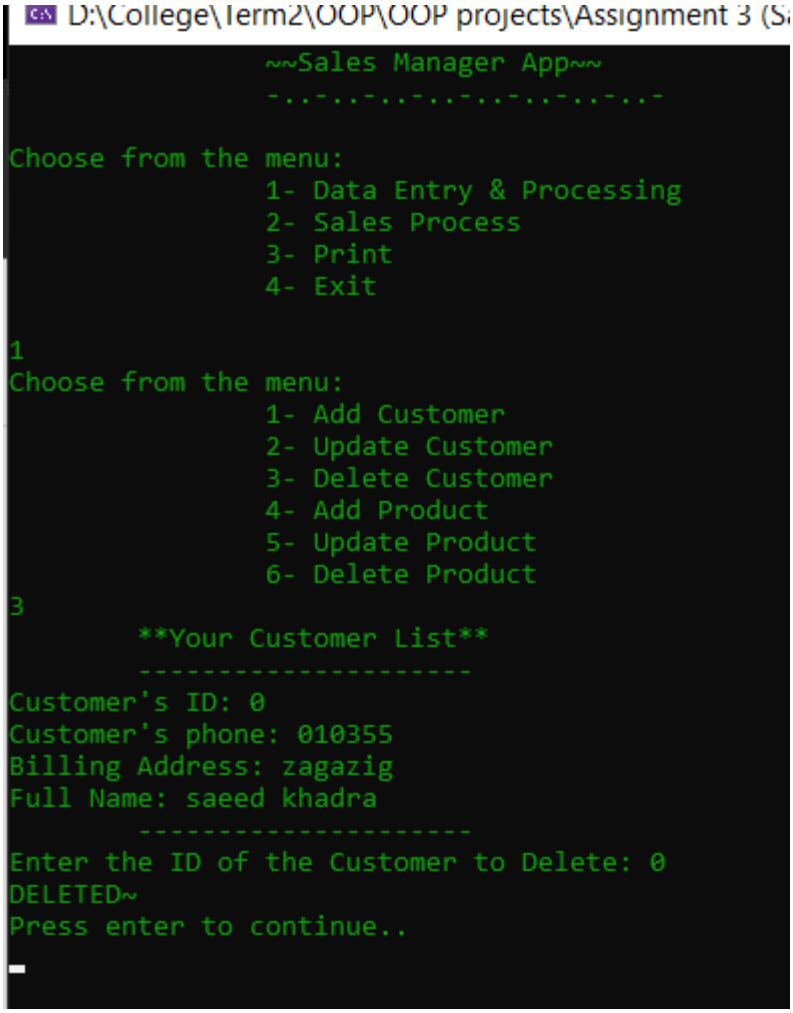
- Adding a customer:



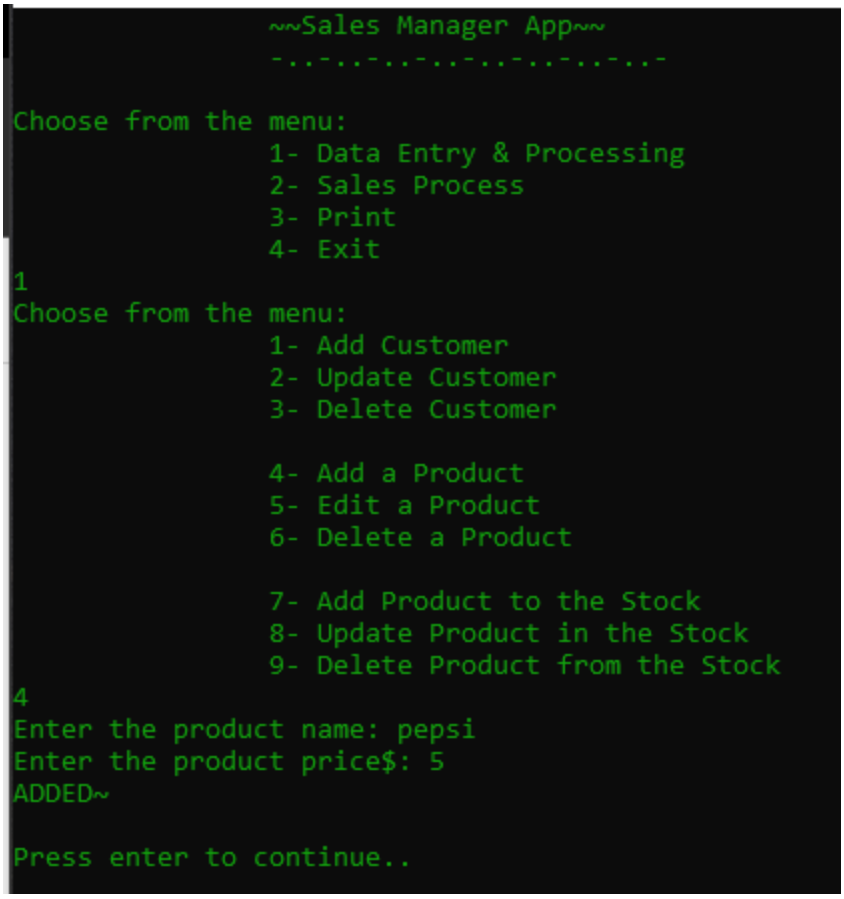
- Update a customer:



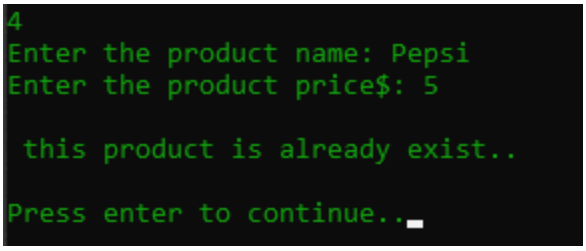
- Delete a customer:



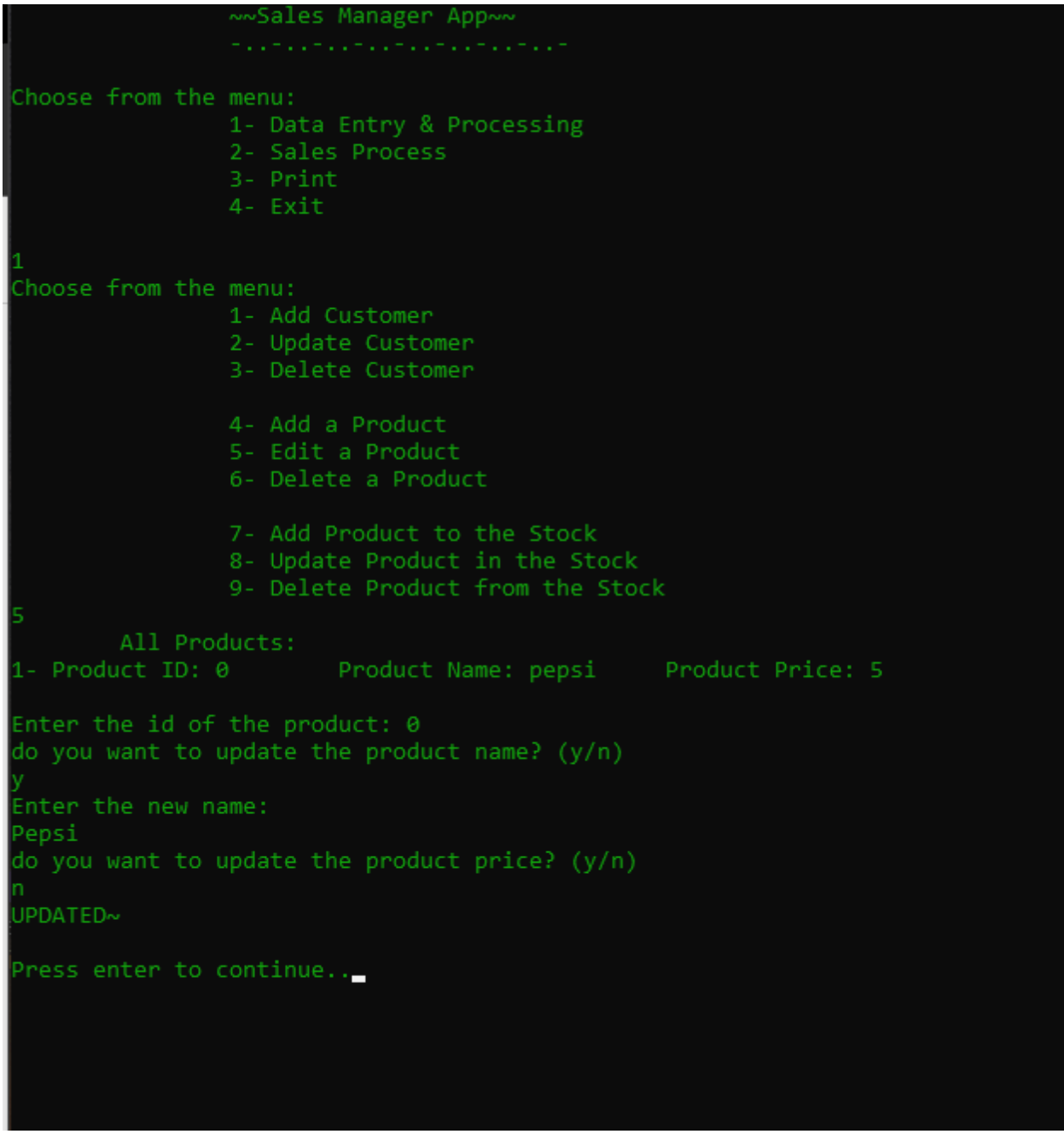
- Add product:



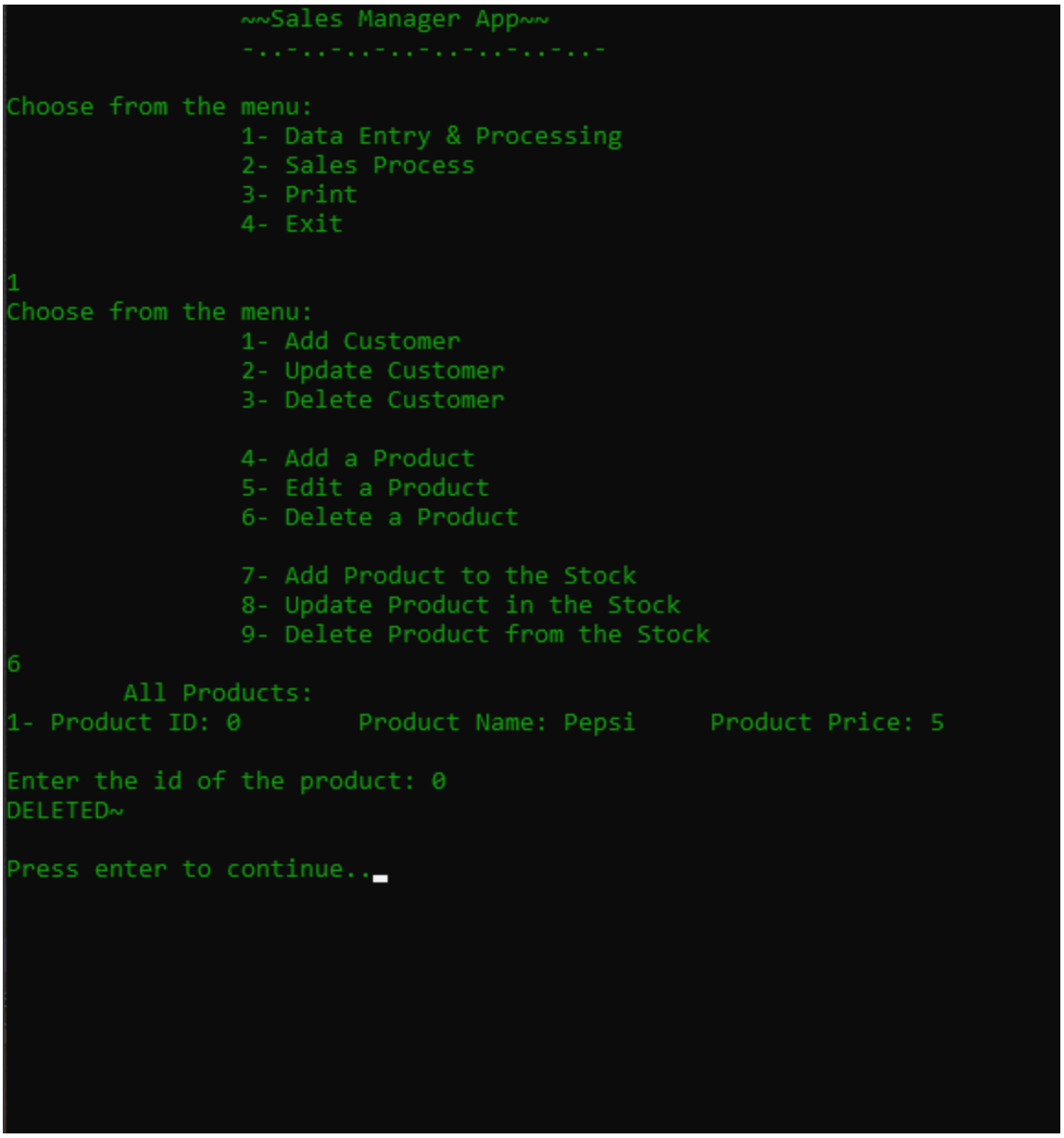
  - With checking if this product is already exist or not..



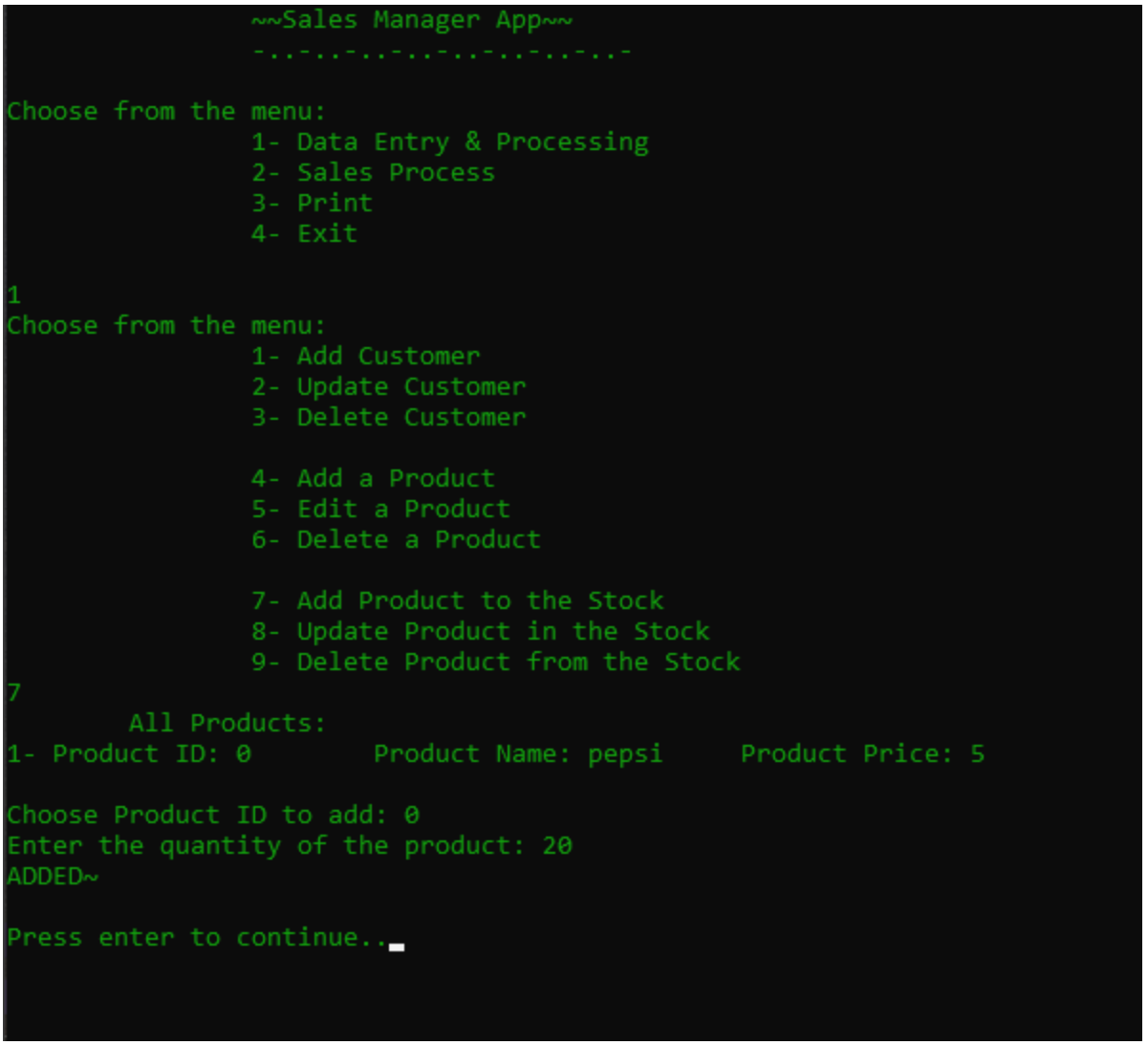
- Edit product:



- Delete a product:



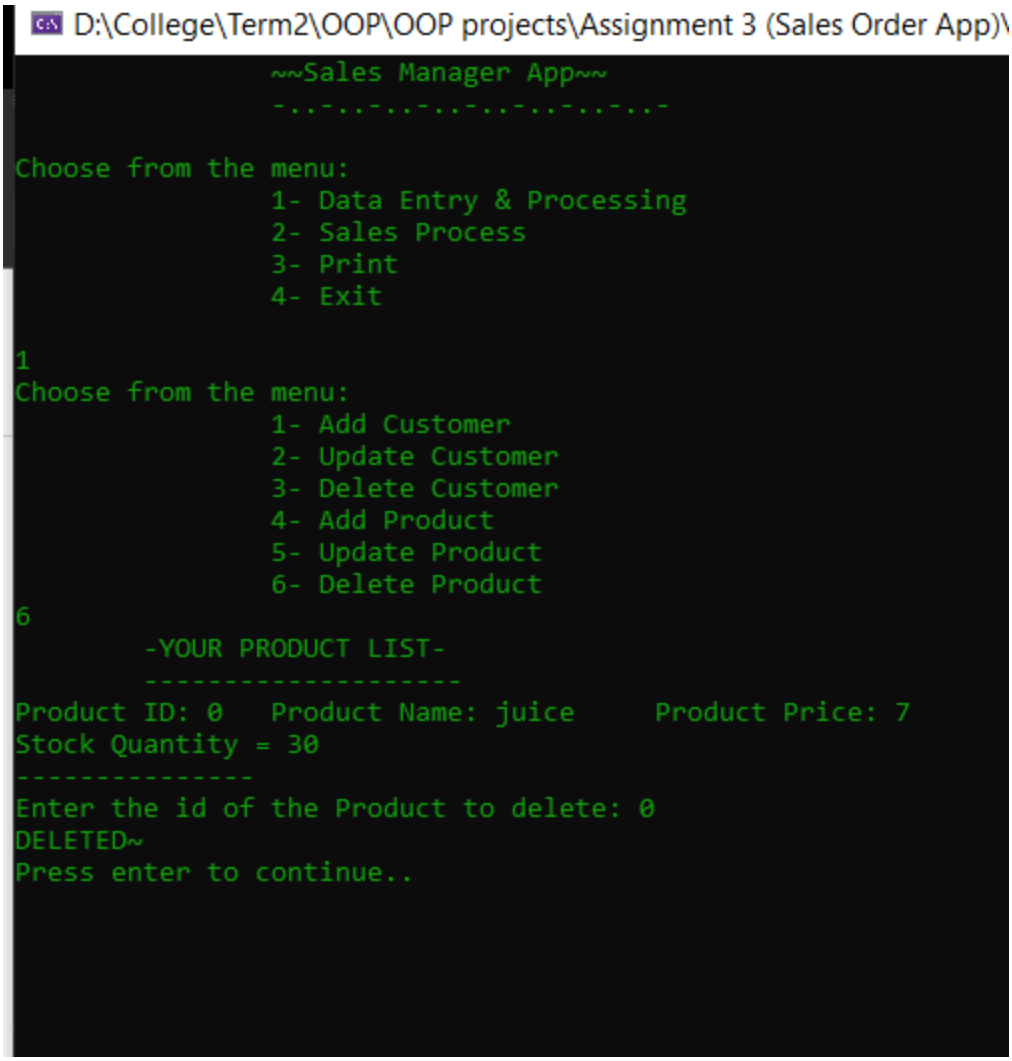
- Adding a product to the stock and its quantity:



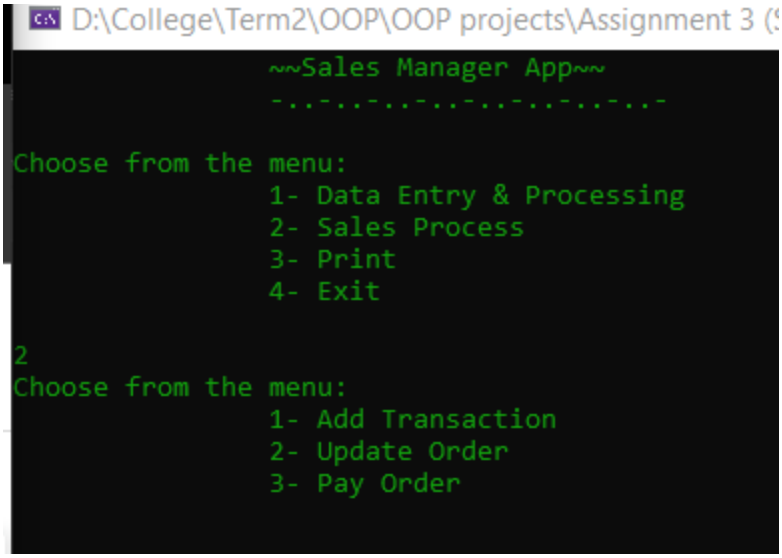
- Update a product:

```
Choose from the menu:
                1- Data Entry & Processing
                2- Sales Process
                3- Print
                4- Exit

1
Choose from the menu:
                1- Add Customer
                2- Update Customer
                3- Delete Customer

                4- Add a Product
                5- Edit a Product
                6- Delete a Product

                7- Add Product to the Stock
                8- Update Product in the Stock
                9- Delete Product from the Stock
8
        -YOUR PRODUCT LIST-
        --------------------
Product ID: 0   Product Name: pepsi     Product Price: 5
Stock Quantity = 30
---------------
Enter the id of the Product to update the quantity: 0
Enter the new quantity: 25
UPDATED~

Press enter to continue..
```

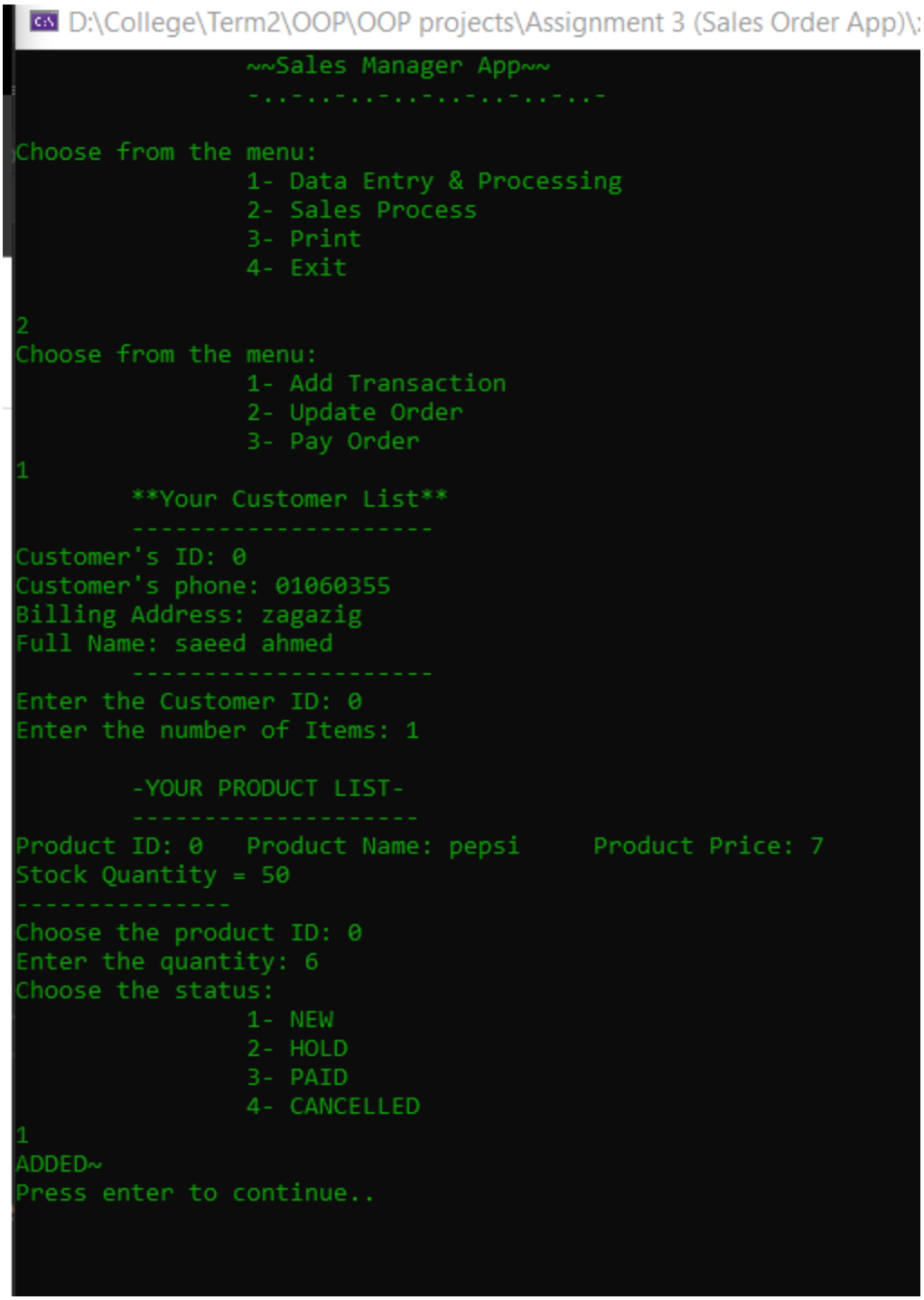
- Delete a product from the stock:



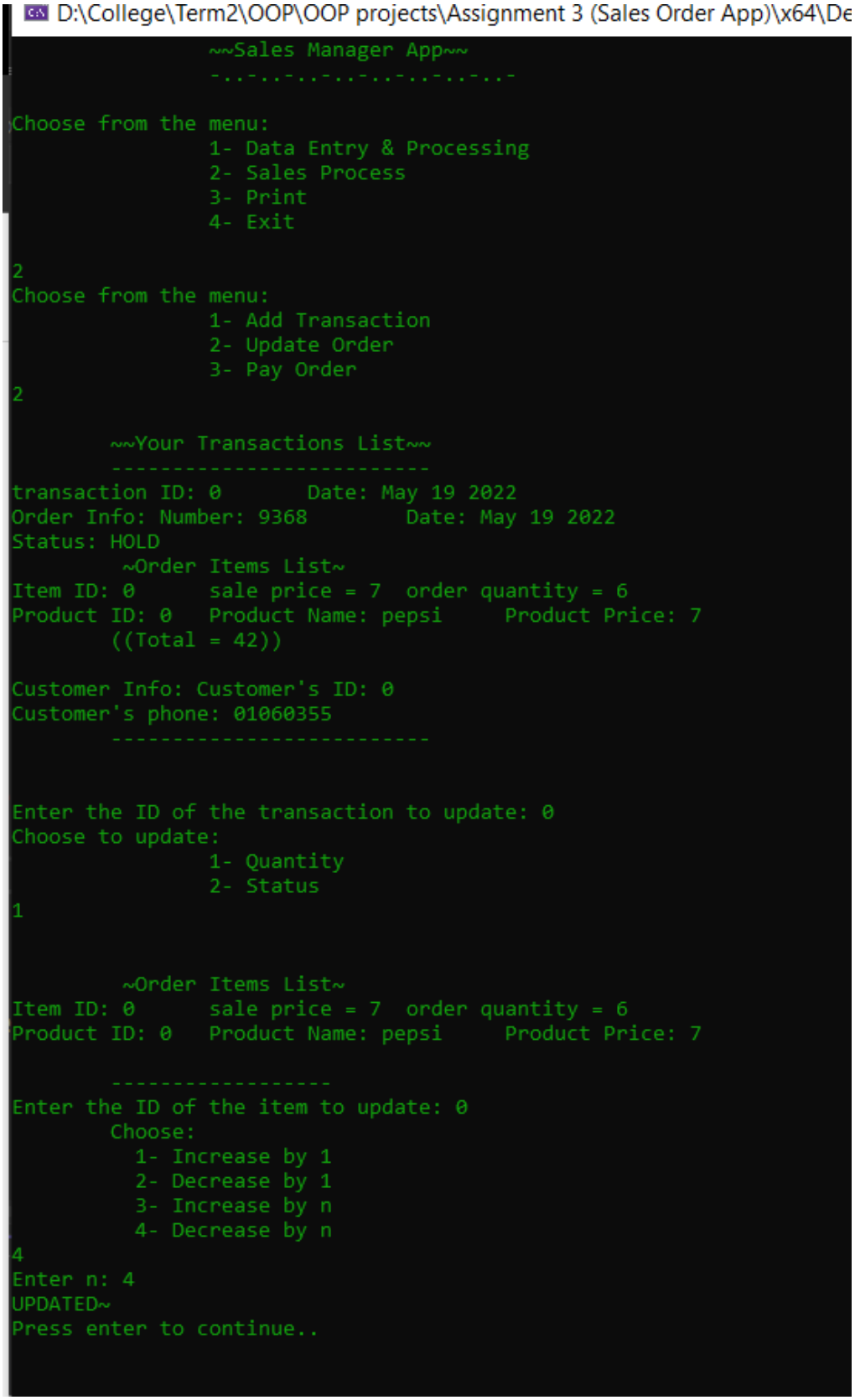
- Sales Process menu:



- Adding a transaction:

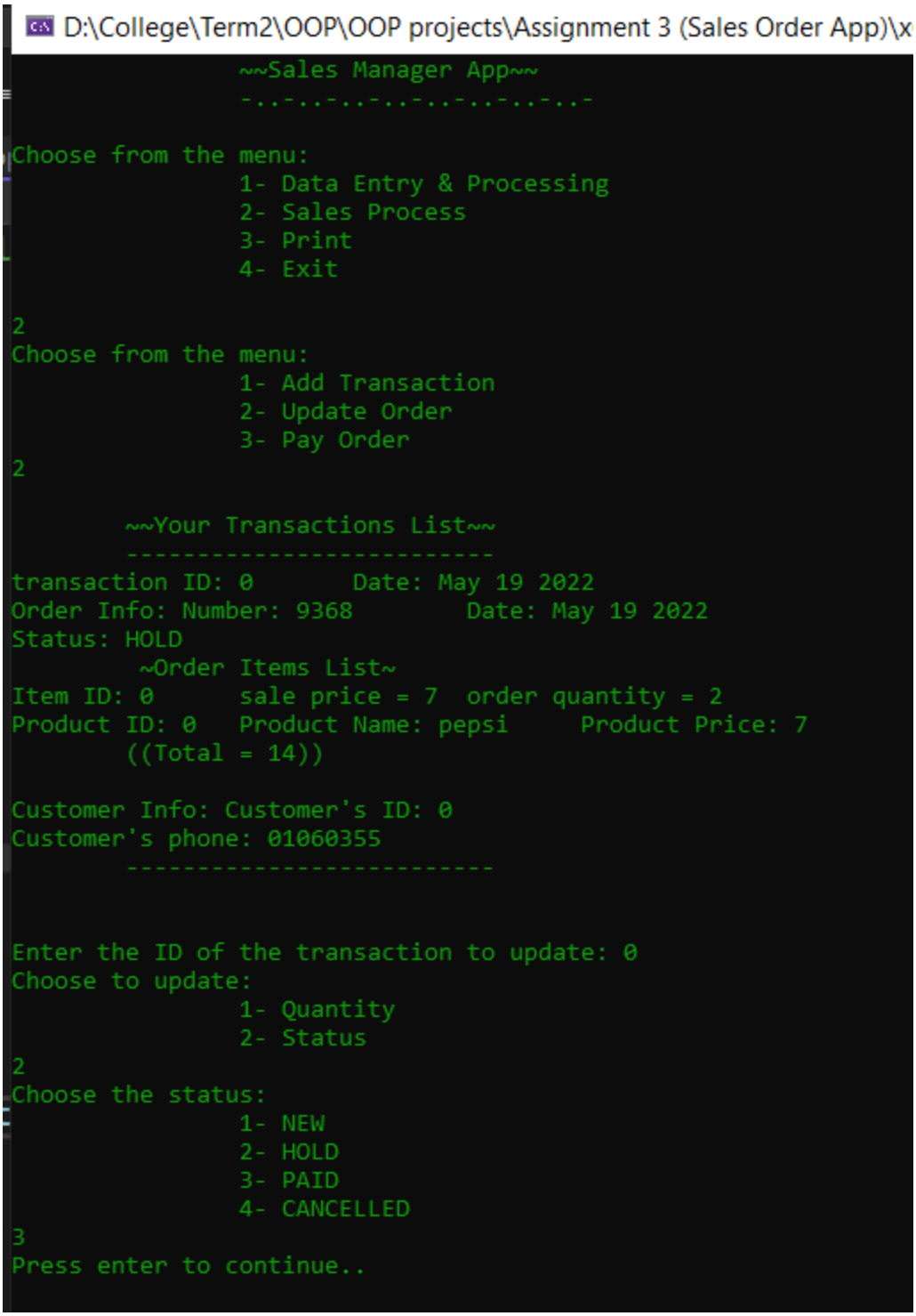


- Update the quantity of an order's order item:



- Using the overloaded mathematical operators.
- Will also update the stock quantity based on the changing in the product quantity.

- Update order status:

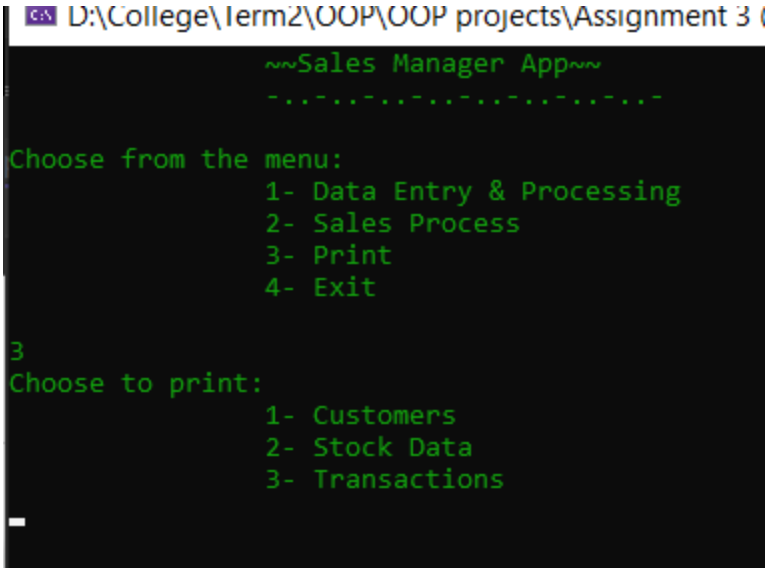


- Pay an order (transaction):

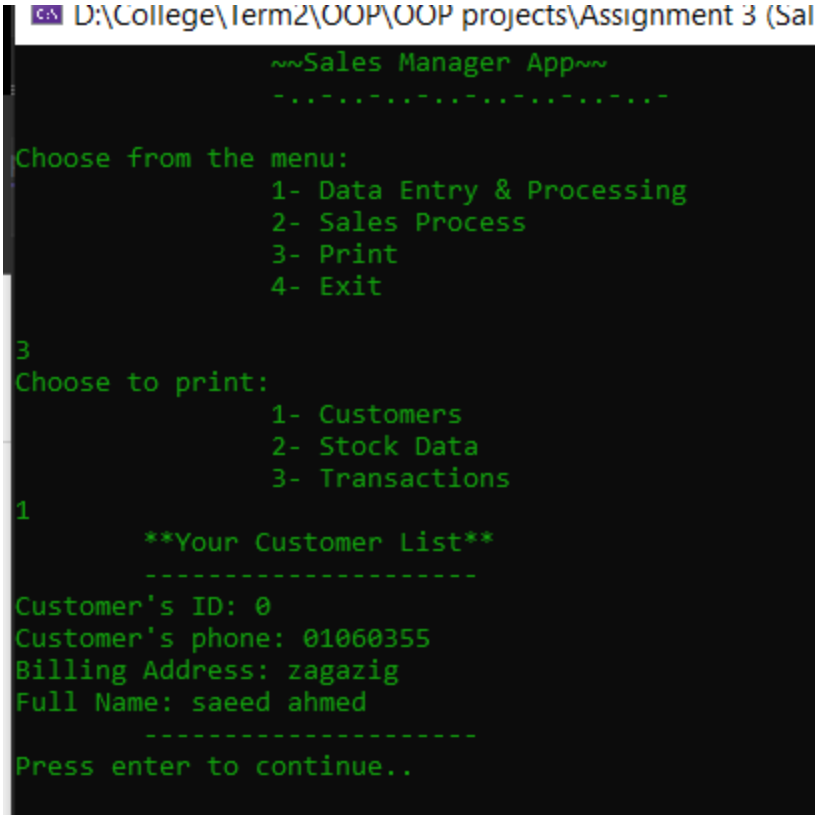
```
                ~~Sales Manager App~~
                -..-..-..-..-..-..-..-..-..-

Choose from the menu:
                1- Data Entry & Processing
                2- Sales Process
                3- Print
                4- Exit

2
Choose from the menu:
                1- Add Transaction
                2- Update Order
                3- Pay Order
3

        ~~Your Transactions List~~
        --------------------------
transaction ID: 0       Date: May 19 2022
Order Info: Number: 9368        Date: May 19 2022
Status: CANCELLED
         ~Order Items List~
Item ID: 0      sale price = 7  order quantity = 2
Product ID: 0   Product Name: pepsi     Product Price: 7
        ((Total = 14))

Customer Info: Customer's ID: 0
Customer's phone: 01060355
        --------------------------

Enter the ID of the transaction to update: 0
Enter the amount to pay: 14
Choose the paying method:
                1- Credit
                2- Cash
                3- Check
2
Enter the cash value: 100
PAID~
Press enter to continue..
```

    - Now will decrease the “total” in the order and will also change the status from NEW to PAID.

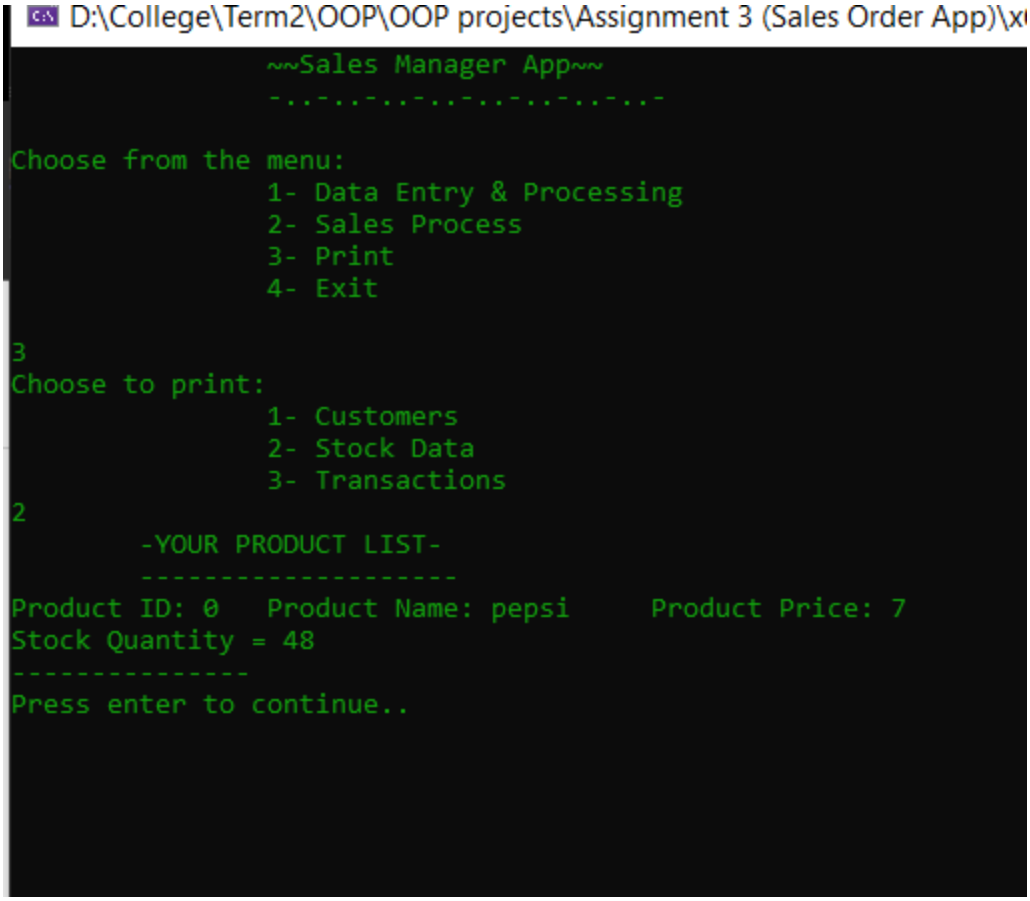
- Print menu:



  - Print customers:



- Print the products in the stock & their quantities:



- Print the transactions:

```
                ~~Sales Manager App~~
                -..-..-..-..-..-..-..-..-

Choose from the menu:
                1- Data Entry & Processing
                2- Sales Process
                3- Print
                4- Exit

3
Choose to print:
                1- Customers
                2- Stock Data
                3- Transactions
3

        ~~Your Transactions List~~
        -------------------------
transaction ID: 0       Date: May 19 2022
Order Info: Number: 9368        Date: May 19 2022
Status: PAID
         ~Order Items List~
Item ID: 0      sale price = 7  order quantity = 2
Product ID: 0   Product Name: pepsi     Product Price: 7
        ((Total = 0))

Customer Info: Customer's ID: 0
Customer's phone: 01060355
        -------------------------

Press enter to continue..
```

  - Which containing the order of the transaction and the customer of it.

- Print the products:

```
                ~~Sales Manager App~~
                -..-..-..-..-..-..-..-..-

Choose from the menu:
                1- Data Entry & Processing
                2- Sales Process
                3- Print
                4- Exit

3
Choose to print:
                1- Customers
                2- Stock Data
                3- Transactions
                4- Products
4
        All Products:
1- Product ID: 0        Product Name: pepsi     Product Price: 5


Press enter to continue..
```

- Finish our journey :)

```
D:\College\Term2\OOP\OOP projects\Assignment 3 (S
                ~~Sales Manager App~~
                -..-..-..-..-..-..-..-..-

Choose from the menu:
                1- Data Entry & Processing
                2- Sales Process
                3- Print
                4- Exit

4
See You Soon :)
```

  - Then will save the final program data in files..