



# OOP3

Sales Order App

---

Your Name: **Saeed Ahmed Khadra**

Your ID: 20912021100427

Your Section: 12

### Application Description:

## Diagrams

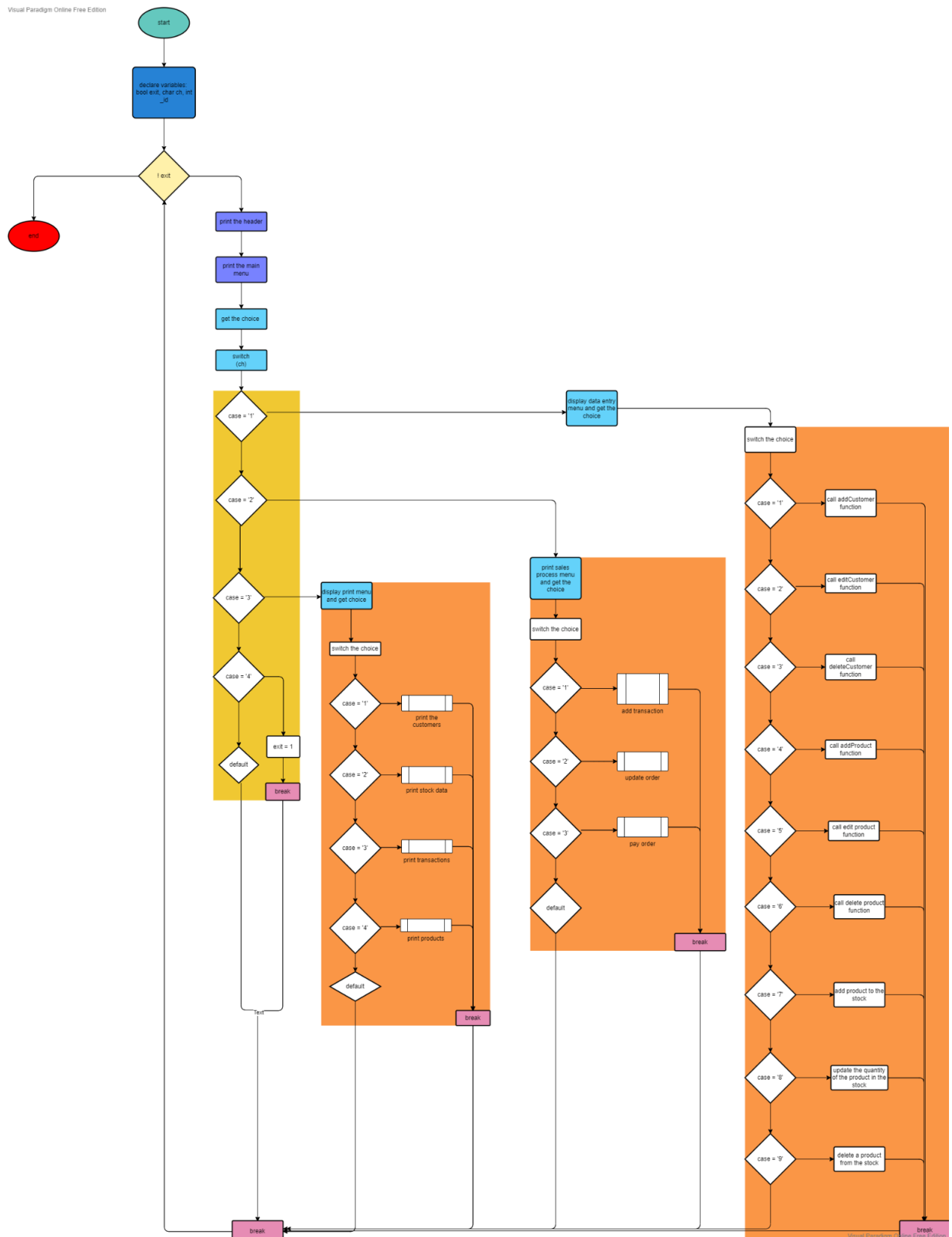
Visual Paradigm Online Free Edition

```
classDiagram
    class Company {
        -location : string
        -company_name : string
        +Company(_id : int)
        +print()
        +update()
    }
    class Person {
        -billing_address : string
        -full_name : string
        +Person(_id : int)
        +print()
        +update()
    }
    class Customers {
        -count : int
        -customerList : vector<Customer*>
        +Customers()
        +- Customers()
        +addCustomer()
        +editCustomer()
        +deleteCustomer()
        +printAllCustomers()
        +getCustomersNum() : int
        +getCustomerPtr(_id : int) : Customer*
    }
    class Order {
        -countOrderItems : int
        -os : orderStatus
        -orderItems : vector<OrderItem>
        -number : int
        -total : double
        -date : string
        -customerPtr : Customer*
        +Order()
        +Order (Customer* ptr)
        +Order (Order& o)
        +addItems(stock : Stock&)
        +printOrderItems()
        +updateStatus(status : int)
        +setStatus(status : int)
        +updateItemQuant(stock : Stock&)
        +updateTotal(pai_amount)
        +getTotal() : double
        +operator<< (input : istream&, o : Order&) : istream&
        +operator<< (output : ostream&, o : Customer) : ostream&
    }
    class Product {
        -id : int
        -name : string
        -price : double
        +Product(Product& p)
        +Product(_id : int)
        +setID (newID : int)
        +setName(newName : string)
        +setPrice(newPrice : double)
        +getPrice() : double
        +getID() : int
        +getName() : string
        +update()
        +operator<< (input : istream&, p : Product&) : istream&
        +operator<< (output : ostream&, p : Product&) : ostream&
    }
    class Stock {
        -count : int
        -stockProducts : vector<Product>
        -productQuant : vector<int>
        +Stock()
        +findProduct(_name : string) : bool
        +addProduct(products : vector<Product*>)
        +updateProduct(_id : int)
        +deleteProduct(_id : int)
        +updateQuant(_id : int, token : int)
        +setQuant(_id : int, new_quant : int)
        +getQuantity(_id : int) : int
        +getProductsCount() : int
        +getProductPrice(_id : int) : double
        +getProduct(_id : int) : Product*
        +print()
        +operator<< (output : ostream&, s : Stock&) : ostream&
    }
    class Credit {
        +number : string
        +type : string
        +date : string
        +Credit(_amount : double)
    }
    class Cash {
        +cashValue : double
        +Cash(_amount : double)
    }
    class Check {
        +name : string
        +bankID : string
        +Check(_amount : double)
    }
    class Payment {
        -paidDate : string
        -amount : double
        +Payment(_amount : double)
        +pay() : double
    }
    class Transaction {
        -id : int
        -transactionDate : string
        -order : Order
        +Transaction(_id : int, o : Order&)
        +Transaction(tr : Transaction&)
        +addOrderItems(stock : Stock&)
        +updateOrderStatus(status : int)
        +updateOrderQuantities(stock : Stock&)
        +payOrder(p : Payment&)
        +setOrderStatus(status : int)
        +getOrderTotal() : double
        +operator<< (output : ostream&, t : Transaction) : ostream&
    }
    class OrderItem {
        -id : int
        -salePrice : double
        -orderQuantity : int
        -product : Product*
        +OrderItem(_id : int, sale_price : double = 0, quant : int = 0, ptr : Product* = NULL)
        +getPrice() : double
        +getItemQuant() : int
        +getProduct() : Product*
        +operator++(int) : OrderItem
        +operator--(int) : OrderItem
        +operator+=(n : int) : OrderItem
        +operator-=(n : int) : OrderItem
        +operator<< (output : ostream&, o : OrderItem) : ostream&
    }
    class EnumOrderStatus {
        -NEW : int = 0
        -HOLD : int = 1
        -PAID : int = 2
        -CANCELLED : int = 3
    }
    Company --|> Person
    Customers --> Order
    Order --> Product
    Order --> Stock
    Order --> EnumOrderStatus
    Order --> Credit
    Order --> Cash
    Order --> Check
    Order --> Payment
    Order --> Transaction
    Order --> OrderItem
    Credit --|> Cash
    Credit --|> Check
    Credit ..> Payment : use
    Transaction ..> Payment : use
    Transaction ..> OrderItem : use
```

Visual Paradigm Online Free Edition

## 2. Flow Diagram

Visual Paradigm Online Free Edition



## 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 sold).	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

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

## 5. Customer

	<b>Function Name</b>	<b>Description</b>	<b>Input</b>	<b>Output</b>
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

	<b>Function Name</b>	<b>Description</b>	<b>Input</b>	<b>Output</b>
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	Constructor 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:

 D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\x64\De

```

~Sales Manager App~
~.....~

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

```

- Data entry menu:

```

~Sales Manager App~
~.....~

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

```

- D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales (

```

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

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 Product
5- Update Product
6- Delete Product

1
Choose the type of the Customer:
1- person
2- company

1
Enter the Customer's phone: 01060355869
Enter the Billing Address: zagazig
Enter the Full Name: saeed ahmed
ADDED~
Press enter to continue..

```

- Update a customer:

```

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

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 Product
5- Update Product
6- Delete Product

2
    **Your Customer List**
    -----
Customer's ID: 0
Customer's phone: 01060355869
Billing Address: zagazig
Full Name: saeed ahmed
    -----

Enter the ID of the Customer to Edit: 0
Enter the new Phone: 010355
Enter the new Billing Address: zagazig
Enter the new Full Name: saeed khadra
UPDATED~
Press enter to continue..
-

```

- Delete a customer:

 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

1
Choose from the menu:
1- Add Customer
2- Update Customer
3- Delete Customer
4- Add Product
5- Update Product
6- Delete Product

3

    **Your Customer List**
    -----
Customer's ID: 0
Customer's phone: 010355
Billing Address: zagazig
Full Name: saeed khadra
    -----
Enter the ID of the Customer to Delete: 0
DELETED~
Press enter to continue..
_
```

- Add product:

```

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

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

4
Enter the product name: pepsi
Enter the product price$: 5
ADDED~

Press enter to continue..

```

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

```

4
Enter the product name: Pepsi
Enter the product price$: 5

this product is already exist..

Press enter to continue..

```

- Edit product:

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

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

5
    All Products:
1- Product ID: 0      Product Name: pepsi      Product Price: 5

Enter the id of the product: 0
do you want to update the product name? (y/n)
y
Enter the new name:
Pepsi
do you want to update the product price? (y/n)
n
UPDATED~

Press enter to continue.._
```



- Delete a product:

```

~Sales Manager App~
~.....~

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

6
All Products:
1- Product ID: 0      Product Name: Pepsi      Product Price: 5

Enter the id of the product: 0
DELETED~

Press enter to continue.._

```

- Adding a product to the stock and its quantity:

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

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

7
      All Products:
1- Product ID: 0      Product Name: pepsi      Product Price: 5

Choose Product ID to add: 0
Enter the quantity of the product: 20
ADDED~

Press enter to continue.._
```

- 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:

```

D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\
    ~~~Sales Manager App~~~
    .....

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 Product
    5- Update Product
    6- Delete Product

6
    -YOUR PRODUCT LIST-
    -----
Product ID: 0   Product Name: juice   Product Price: 7
Stock Quantity = 30
-----
Enter the id of the Product to delete: 0
DELETED~
Press enter to continue..
  
```

- Sales Process menu:

```
D:\College\Term2\OOP\OOP projects\Assignment 3 (C
~~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
```

- Adding a transaction:

```

D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\:
~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

1
    **Your Customer List**
    -----
Customer's ID: 0
Customer's phone: 01060355
Billing Address: zagazig
Full Name: saeed ahmed
    -----
Enter the Customer ID: 0
Enter the number of Items: 1

    -YOUR PRODUCT LIST-
    -----
Product ID: 0   Product Name: pepsi   Product Price: 7
Stock Quantity = 50
    -----
Choose the product ID: 0
Enter the quantity: 6
Choose the status:
1- NEW
2- HOLD
3- PAID
4- CANCELLED

1
ADDED~
Press enter to continue..

```

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

📁 D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\x64\De

```

      ~~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

2

      ~~Your Transactions List~~
      ~~~~~
transaction ID: 0      Date: May 19 2022
Order Info: Number: 9368      Date: May 19 2022
Status: HOLD
      ~Order Items List~
Item ID: 0      sale price = 7      order quantity = 6
Product ID: 0      Product Name: pepsi      Product Price: 7
      ((Total = 42))

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

Enter the ID of the transaction to update: 0
Choose to update:
      1- Quantity
      2- Status

1

      ~Order Items List~
Item ID: 0      sale price = 7      order quantity = 6
Product ID: 0      Product Name: pepsi      Product Price: 7
      ~~~~~

Enter the ID of the item to update: 0
Choose:
      1- Increase by 1
      2- Decrease by 1
      3- Increase by n
      4- Decrease by n

4
Enter n: 4
UPDATED~
Press enter to continue..

```

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

```

C:\> D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\x

~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

2

~Your Transactions List~
-----
transaction ID: 0      Date: May 19 2022
Order Info: Number: 9368      Date: May 19 2022
Status: HOLD
~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
Choose to update:
1- Quantity
2- Status

2
Choose the status:
1- NEW
2- HOLD
3- PAID
4- CANCELLED

3
Press enter to continue..

```



- Pay an order (transaction):

```

D:\College\term2\OOP\OOP projects\Assignment 3 (Sales Order App)\x04\De
~~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:

```

C:\ D:\College\Term2\OOP\OOP projects\Assignment 3 (
    ~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

```

- Print customers:

```

C:\ D:\College\Term2\OOP\OOP projects\Assignment 3 (Sal
    ~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

1
    **Your Customer List**
    -----
Customer's ID: 0
Customer's phone: 01060355
Billing Address: zagazig
Full Name: saeed ahmed
    -----
Press enter to continue..

```

- Print the products in the stock & their quantities:

C:\ D:\College\Term2\OOP\OOP projects\Assignment 3 (Sales Order App)\xi

```
~~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

2
    -YOUR PRODUCT LIST-
    -----
Product ID: 0   Product Name: pepsi   Product Price: 7
Stock Quantity = 48
-----
Press enter to continue..
```

- 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 :)

```

C:\> 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..