

**Zagazig University (FCI\_ZU).**

**Faculty Of Computer Science and Artificial Intelligence**

Sales Order System

Object Oriented Programming

(Assignment 3)

Under Supervisor:

* Dr. Mahmoud A. Mahdi

Programmed by:

* Sherif Muhammad Muhammad Abdulsalam.
* Set Number: 1315
* Student Number: 20912021100785
* Group: A2
* Section: 13

**Application Description**

**Sales Order System is a C++Program that helps control the process between a customer placing an order and the product. The sales order confirms the terms of a transaction between a buyer and seller. The seller generates the order, often in response to a purchase order. The seller may send this document to the customer or rely on it solely for internal use. The sales order details the quantity, price, delivery time frame, and more.**

**Data Structure Used**

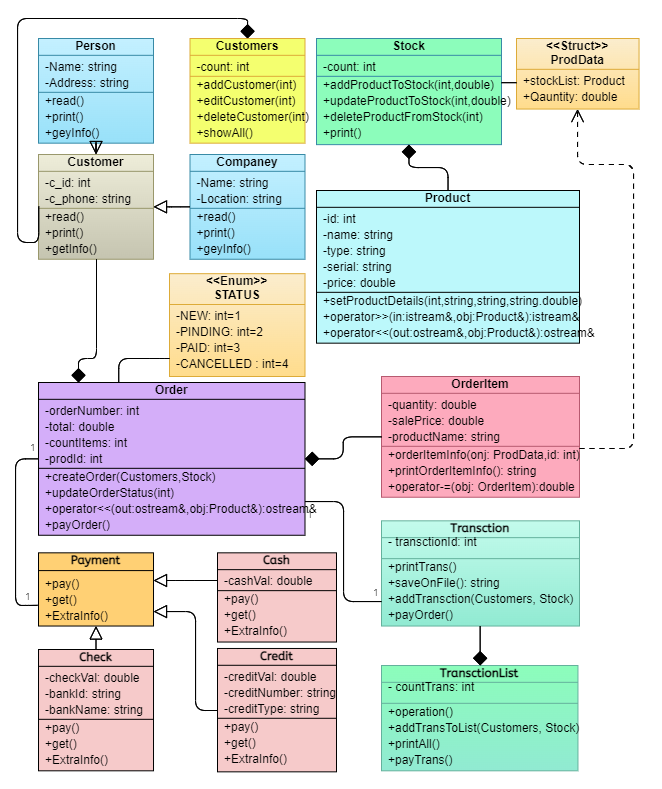
**Arrays has a variety of datatypes like built in datatypes (char, int, string…. etc.) and other are user defined data types (Customer, Product, and other). These data were collected to provide the best solution for collecting data together to be easily reached and used. To give the best performance, more speed and save the memory we choose dynamic array to be used due to its performance and because it can fit automatically after insert or deleting objects or data from the array.**

**Enumeration << Enum >> was used to set some constants for order Status like (NEW, PINDING, PAID, CANCELLED) to know how to deal with orders.**

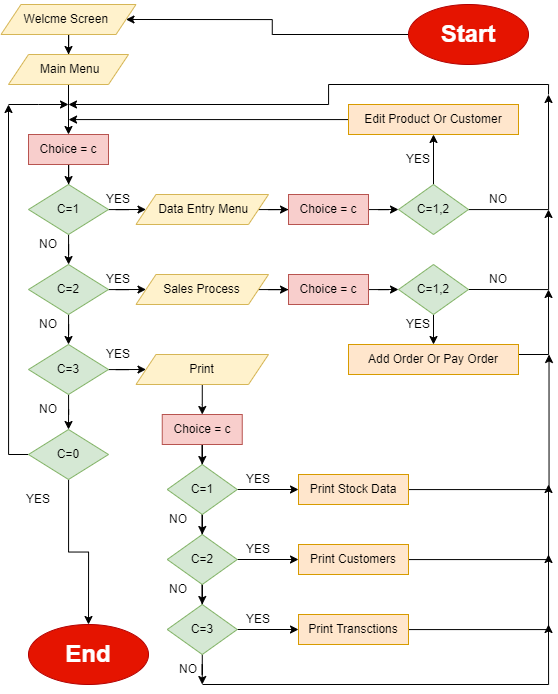
**Structure << Struct >> Was Used to Collect the full data for each product in addition to adding quantity in the stock for each one of them.**

Diagrams

1. **UML Diagram**



1. **Flow Diagram**

****

Classes & Functions

1. **Struct Class**

**The Job of This Structure class Is to Hold the product and Its Quantity**

1. **Stock Class**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function Name | Description | Input |
|  | **Add Product to Stock** | **It takes products from product factory and added it to stock** | **Product ID And Quantity** |
|  | **Update Product to Stock** | **It can edit an existing product in the stock** | **Product ID And New Quantity** |
|  | **Delete Product to Stock** | **It can delete an existing product in the stock** | **Product ID** |
|  | **print** | **It Prints All Existed Products in the stock** | **NO** |
|  | **Get Product Data** | **It gets each product in the stock with its quantity** | **NO** |

1. **Product Class**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function Name | Description | Input |
|  | **Set Product Details** | **It set All Attributes of each product from id, name, serial number, type and price** | **Product Id, Name, Serial,**  **type, price** |
|  | **Operators <<, >>** | **Handling C++ Operators to Work with Objects from Class Product** | **Overloaded In C++** |

1. **Order Item Class**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function Name | Description | Input |
|  | **Update Stock Product Quantity** | **It updates the quantity of its product in the stock** | **NO** |
|  | **Order Item Info** | **This Method is responsible for setting the information for each order item from stock class** | **Point to the structure to the stock and the product id** |
|  | **Print Order Item Info** | **This Function Print the sale price, name and the quantity** | **NO** |
|  | **Operators -=** | **Handling C++ Operators to Work with Objects from Class Order Item** | **Overloaded In C++** |

1. **Customer Class**

|  |  |  |
| --- | --- | --- |
|  | Function Name | Description |
|  | **Read** | **It takes the customer DATA and store them in private attributes** |
|  | **Print** | **Print the customer Data in C++ Console Screen** |
|  | **Get Info** | **Print the customer Data in C++ File Stream** |

1. **Person, Company inherited Classes Form Customer Class**

|  |  |  |
| --- | --- | --- |
|  | Function Name | Description |
|  | **Read** | **It overrides on the on the function read Which Was Inherited from Class Customer and Added To it Person or company Private Data Like Phone or Address or location.** |
|  | **Print** | **Print the Person/company Data in C++ Console Screen** |
|  | **Get Info** | **Print the Person/company Data in C++ File Stream** |

1. **Customers Class**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function Name | Description | Input |
|  | **Add New Customer** | **adds new Customer to customer list** | **Customer Id** |
|  | **Delete Customer** | **It can delete an existing Customer in the customer list** | **Customer Id** |
|  | **Edit Customer** | **It can edit an existing Customer in the customer list** | **Customer Id** |
|  | **Show All Customers** | **Print customer list** | **NO** |
|  | **Get customer list** | **Get customer list to use It in other classes in the program** | **NO** |

1. **Enum Class**

**It contains some contestants represent order status (NEW, PINDING, PAID, CANCELLD)**

1. **Order Class**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function Name | Description | Input |
|  | **Create Order** | **This Function display all customers and choose one of them then display all products int the stock to create order for this customer** | **Pointer to class customer and pointer to class Stock** |
|  | **Update Order Status** | **It responsible for changing the order status if paid or pending or new even cancelled** | **Integer number to represent status** |
|  | **Edit Customer** | **It can edit an existing Customer in the customer list** | **Customer Id** |
|  | **pay** | **This Function can calculate the total amount of order and choose one way to pay from cash or credit or check.** | **NO** |
|  | **Operators <<** | **Handling C++ Operators to Work with Objects from Class Order Item** | **Overloaded In C++** |

1. **Payment Class**

**This class contain three virtual pure function to set the total amount paid and one to get it and the another one to set Extra Information.**

1. **Cash, Credit, Check inherited Classes Form Payment Class**

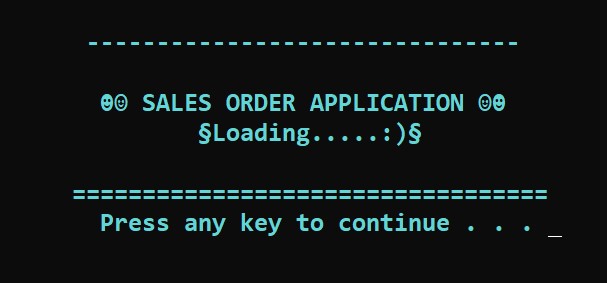
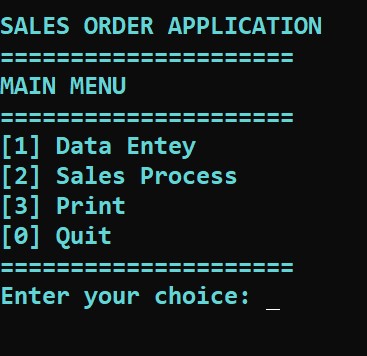
|  |  |  |
| --- | --- | --- |
|  | Function Name | Description |
|  | **pay** | **This Function This Function Asks customer to pay for chosen order and choose one way to pay** |
|  | **get** | **Return to total amount paid from customer** |
|  | **Extra Info** | **It prints Extra Information like card Number, Type in credit Way or bank name, id in check way** |

1. **Transaction**

|  |  |  |
| --- | --- | --- |
|  | Function Name | Description |
|  | **Add Transaction** | **This function used to create order and its date and status** |
|  | **Print transaction** | **display all Transaction Which was created successfully.** |
|  | **Pay Transaction** | **You can choose one transaction to pay.** |

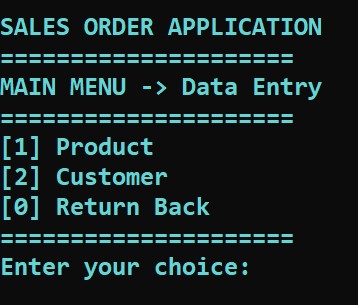
Demo (Scenario Case & Screen Shots)

**After the welcome screen appears, the system starts with The Main Menu of Program.**

****

**If You Select choice <<1>>, then the**

**Data Entry Menu Will Appears to detect**

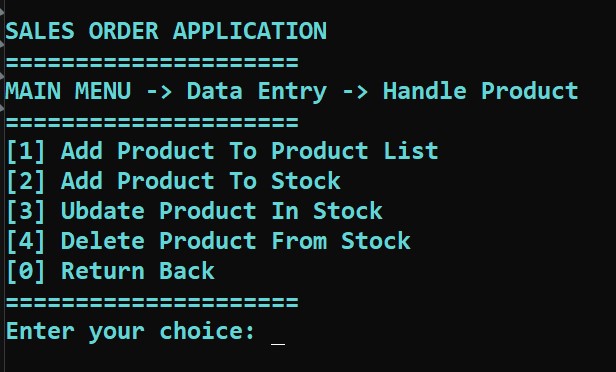
**Which would you want to add…?**

**if you choose << 1>>, the all fetchers**

**of product you can use it from adding**

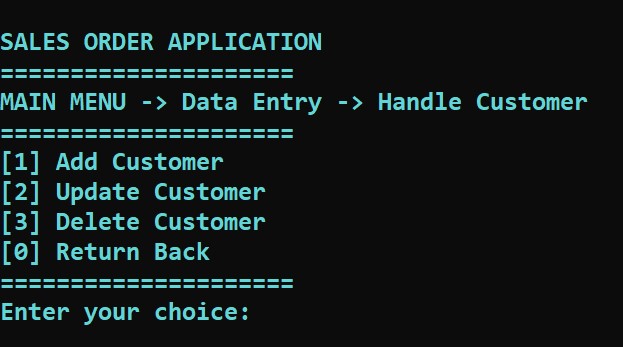
**new product or update an exited product**

**or delete product from the stock.**

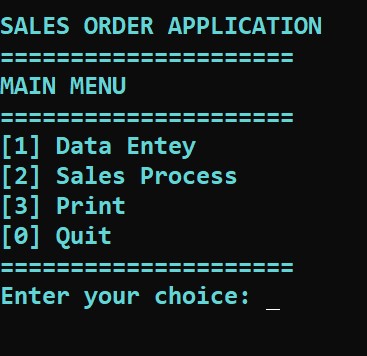
****

**if you select << 0 >> from Handle Product Menu, you will return back to Data Entry Menu.**

**But if you choose << 2 >> you can use the same fetchers of product but this Time on customers.**

****

**If you choose << 0 >> that mean you want to return back to the main menu.**

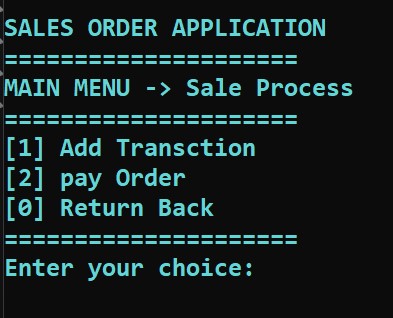
****

**If you input choice << 2>>,**

**The sales process Menu which**

**Contains 3 Fetchers Will Appears**

**And you should select one of them.**

**If you select << 1 >>**

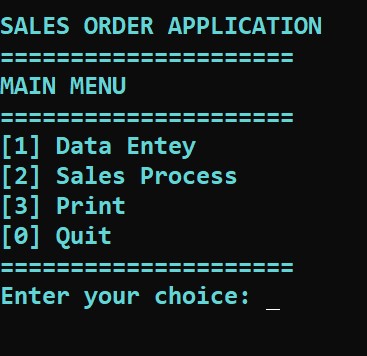
**The program will lead**

**You to the right steps**

**To create a successful**

**Transaction.**

**If << 2 >>**

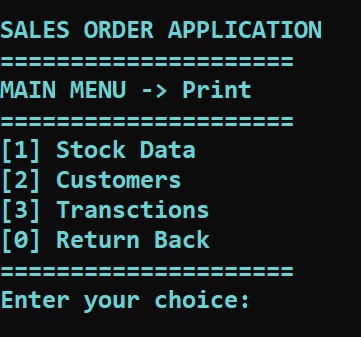
**You will choose only one transaction from transaction list to pay with any way of pays (cash, check, credit).**

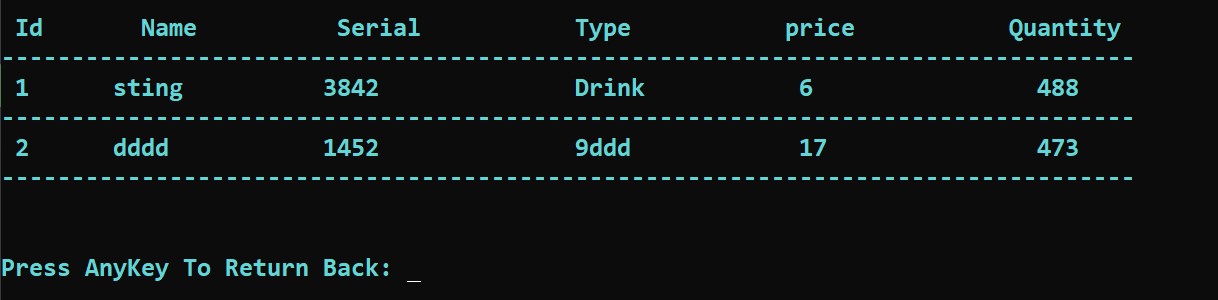
**If << 0 >> return back to the main menu.**

**If you select << 3 >> from start**

**That mean you want Know**

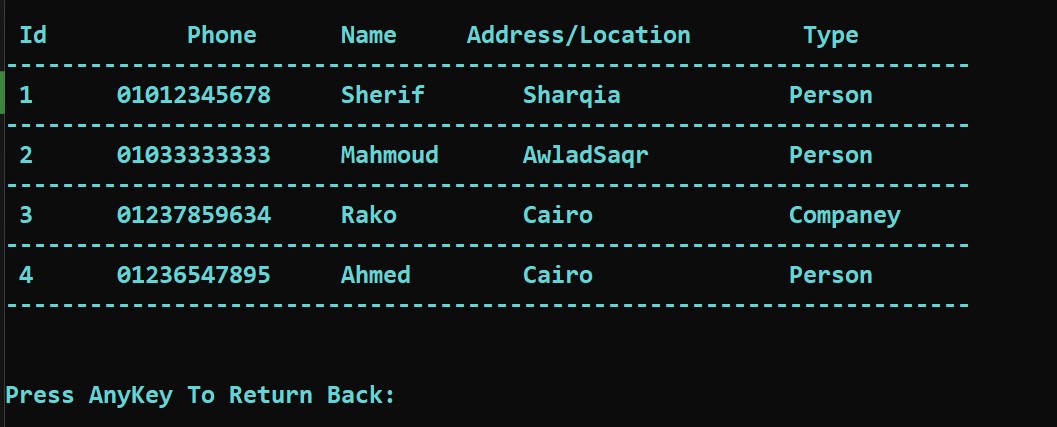
**What do you have???**

****

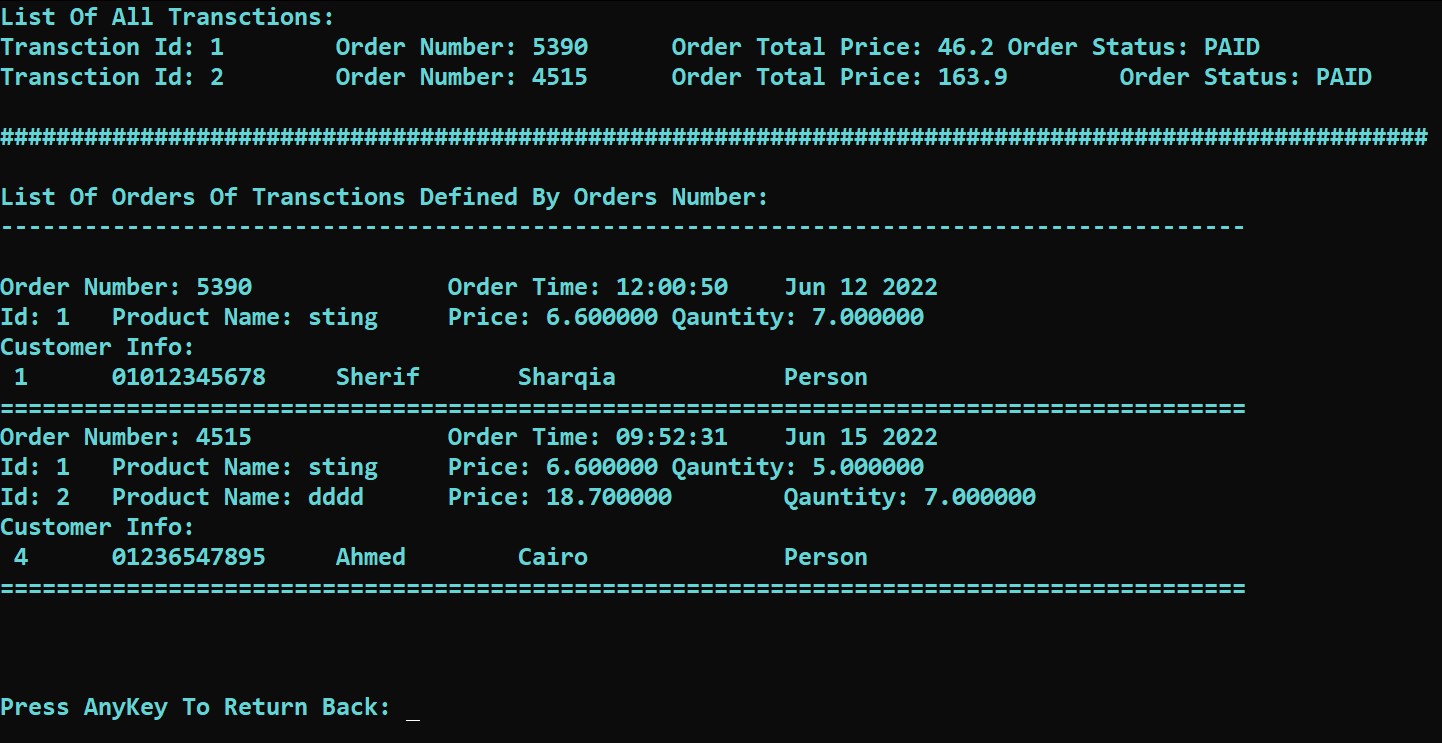
****

**Show Stock Product List**

**Show Customer List**

****

**Show Transaction List**

****