OOP Project (C++)

Pharmacy Management System

- **Objectives:**

1.To construct 7 classes:

Medication

Customer

Address

Date

Pharmacy

Off-the-shelf

Prescription

2. To use setters and getters for data members attributes.

3.To build different types of constructors (default, parametrized, copy).

4. To use composition concepts and principles.

5. To create relations using inheritance.

6.To allocate objects and arrays in the heap memory (DMA).

7.To use the concept of file separation for more accuracy and clarity.

- **Group Members:**

1.Laila Sadideen.

2.Dana Salem.

3.Yousif Saddouq .

4.Khaled Abu Seir.

summary about the parts of each class:

Medication Class:

1. Protected Data Members

id (int): identifier that uniquely identifies each medication.

idCount (static int): variable that is used to track the total number of medication objects created.

name , description, barcode (string): stores the name, description, barcode of the medication

price(double): stores the medication price

quantity(int) : stores the quantity of the medication in stock

exp_date (Date): object as a data member by using composition from class date , it contains the expiry date (day-month-year)

2.Public Data Members

Default constructor : initializes a medication with default values.
ID receive a counter id count assigns a distinct ID to every medication (member list initializer)

Copy constructor : creates a new Medication object by copying the values from an existing obj.
increments ID count counter and gives a unique identifier.

parameterized constructor : takes parameters to create a medication with these values.
increments ID count counter and gives a unique identifier.

Destructor : called automatically when a medication object goes out of scope.

set name : sets the name of the medication. it
defaults to "no name", If an empty string is provided.

set description: sets the description of the medication. it defaults to "no description", If an empty string is provided .

set exp date: sets the expiry date of the medication.

set price: sets the price of the medication.it defaults to 0, If a negative value is provided.

set quantity: sets the quantity of the medication in stock. it defaults to 0, If a negative value is provided.

set barcode: Sets the barcode of the medication.it defaults to "000000000000", If the provided barcode length is not 12.

get name: returns the name of the medication.

get description: returns the description of the medication.

get exp date: returns the expiry date of the medication as a Date object.

get price: returns the price of the medication.

get quantity : returns the quantity of the medication in stock.

get barcode: returns the barcode of the medication.

get ID: returns ID of the of the medication.

Display Medication Info : print the detailed information about the medication, its name, description, price, quantity in stock, expiry date, and barcode.

Customer Class

1. Private Data Members **customer name (string) :**

stores the name of customer.

customer ID (int) : identifier that uniquely identifies each customer.

Address (Address):object as a data member by using composition from class address , it contains the customer address details

number of customer(static int): variable that is used to track how many Customer objects have been created

2. Public Data Members

Default Constructor : initializes a customer with default values.
Customer ID receive a counter Number Of Customers assigns a distinct ID to every customer (member list initializer)

Parameterized Constructor : takes a name (n) and an Address object (ad) as parameters to create a customer obj with these values, increments ID Count counter and gives a unique identifier.

Copy Constructor : creates new Customer object by copying the values from existing obj

Destructor : called automatically when a customer object goes out of scope.

Name Setter :set the customer's name, the default value is 'no name' if an empty string is provided.

Address Setter : sets the customer's address .

Name Getter : returns the customer's name .

Address Getter : returns customer's address.

ID Getter : returns ID of the customer.

Display Customer Info: print the detailed information about the customer, which includes their name, ID, address. Displays the address details using the print function of the address class.

Date Class

1.Private Data Members

Day (int): stores the day part of the date

Month (int): stores the month part of the date

Year (int): stores the year part of the date

2.Public Data Members

Defaulted Constructor : initializes a date object if no values are provide with default values for day->1 , month -> 1 ,and year->2023

Day Setter: Sets the day part of the date. if the provided value is less than and greater than 31 it defaults to 1.

Month Setter : sets the month part of the date. if the provided value is less than 1 and greater than 12 it defaults to 1.

Year Setter : sets the year part of the date. if the provided value is not greater than or equal 2000, it defaults to 2000.

Day Getter: returns the day part of the date.

Month Getter :returns the month part of the date.

Year Getter: returns the year part of the date.

Display Date: the complete date is printed in the format of "day-month-year."

Address class

1.Private Data Members

Email (string): Stores the email address.

City (string): Stores the city name.

MobileNO (string): Stores the mobile number.

StreetName (string): Stores the street name.

2.Public Data Members

Defaulted Constructor : initializes an address object with default or provided values for email, city, mobile number, and street name.

set Email : sets the email address. it defaults to "xoxo@gmail.com",If an empty string is provided. **set City:** sets the city name. it defaults to "unknown", If an empty string is provided.

set MobileNO: sets the mobile number. it defaults to "+962xxxxxxxxx", If an empty string is provided. **set Street Name:** sets the street name.it defaults to "unknown", If an empty string is provided. **get Email:** returns the email address. **get City:** returns the city name.

get MobileNo: returns the mobile number.

get Street name: returns the street name.

Print Address Info : print the detailed information about the address, email, city, mobile number, and street name.

Prescription Class

1.Private Data Members

FDANumber (int): stores the FDA number

ApprovalDate (Date): stores the date on which the prescription received approval.

2.Public Data Members

Default Constructor: initializes a prescription with default values for medication FDA number, and approval date.

Parameterized Constructor : takes parameters to create a prescription with these values for medication FDA number, and approval date.

set FDANumber: sets the FDA number ,sets the FDA number to 0 If a negative value is provided **set ApprovalDate:** sets the approval date of the prescription.

get FDANumber: returns the FDA number

get ApprovalDate: returns the approval date of the prescription as a Date object.

Display Prescription Info: calling print medication function which print the detailed information about the medication (using scope)
Print the detailed information about the prescription FDAnumber,
and approval date.

OffTheShelf Class

1.Private Data Members

BOGOF (bool): Indicates whether the medication has a (buy one get one free offer)

OfferEnds (Date): stores the end date of the BOGOF offer.

2. Public Data Members

Default Constructor : initializes an OffTheShelf object with default values for the medication attributes, sets BOGOF to false, and calculates the offer ends date two years from the current date if there is no BOGOF offer, or it ends after 3 months from the current date if there was a BOGOF offer.

Parameterized Constructor : initializes an OffTheShelf object with these values for the medication members, and calculates the offer ends date two years from the current date if there is no BOGOF free ,or it ends after 3 months from the current date if there was a BOGOF offer it.

set BOGOF: sets the BOGOF of the medication if true or false .

set OfferEnds:takes a Date object as a parameter ,sets the end date of the BOGOF offer.

get BOGOF : returns the BOGOF of the medication if true or false.

get OfferEnds: returns the end date of the BOGOF offer as a Date object.

Display OffTheShelf Info: calling print medication function which print the detailed information about the medication (using scope) prints the detailed information about the medication, no BOGOF / yes BOGOF, and offer end date.

Pharmacy Class

1.Private Data Members

id (int): stores the medication id

count_id (int, static): stores the last id given to a medication

instance **name (string):** stores the name of the medication

count_med (int): stores the count of medications in pharmacy

count_cust (int): stores the count of customers in pharmacy

count_off (int): stores the count of off the shelf medications in pharmacy

count_pres (int): stores the count of prescription medications in pharmacy

med (Medication*): a dynamic array that stores the medications in pharmacy

cust (Customer*): a dynamic array that stores the customers in pharmacy **off**

(OffTheShelf*): a dynamic array that stores the off the shelf medications in

pharmacy

pres (Prescription*): a dynamic array that stores the prescription medications in pharmacy

2.Public Data Members

Default Constructor : initializes a pharmacy object if no values are provided with default values for name->"No Name" , count_med-> 0 ,and count_cust->0, count_off->0, count_pres->0, med->NULL, cust->NULL, off->NULL, pres->NULL

Parameterized Constructor: initializes a pharmacy object with the provided name and sets the rest of data members as the default constructor

Copy constructor: initializes a pharmacy object with the values of the object to copy from and does deep copy on the dynamic arrays.

Destructor: deletes the dynamic arrays in heap if they are not NULL

Add_medication(Medication med_obj): takes a medication object and adds it to the **med** array. It resizes the dynamic array on each add by 1.

Add_Prescription(Prescription pres_obj): takes a prescription object and adds it to the **pres** array. It resizes the dynamic array on each add by 1.

Add_OffTheShelf(OffTheShelf off_obj): takes a offtheshelf object and adds it to the **off** array. It resizes the dynamic array on each add by 1.

Remove_medication(int index, int type): takes the index and type of medication to remove. If the type is 1 (Type is Medication), it will remove the medication with index **index** from the **med** array and shifts it to the end. Then resizes the array by -1. If the type is 2 (Type is OffTheShelf), it will remove the offtheshelf medication with index **index** from the **off** array and shifts it to the end. Then resizes the array by -1. If the type is 3 (Type is Prescription), it will remove the prescription medication from the **pres** array and shifts it to the end. Then resizes the array by -1.

Remove_medication(string n, int type): takes the name and type of medication to remove. If the type is 1 (Type is Medication), it will remove the medication with name **n** from the **med** array and shifts it to the end. Then resizes the array by -1. If the type is 2 (Type is OffTheShelf), it will remove the offtheshelf medication with name **n** from the **off** array and shifts it to the end. Then resizes the array by -1. If the type is 3 (Type is Prescription), it will remove the prescription medication with name **n** from the **pres** array and shifts it to the end. Then resizes the array by -1.

Add_Customer(Customer cust_obj): takes a customer object and adds it to the **cust** array. It resizes the dynamic array on each add by 1.

av_med(): prints the medications in the **med**,**off**,**pres** dynamic arrays in the pharmacy.

Print_cust(): prints the customers in the pharmacy.

Set_id(int i): sets the id. It checks whether the sent id is greater than or equal to 0. If it is below zero it will set the id as 0.

Get_id(): returns the id.

Set_med_count(int i): sets the count_med as **i**.

Set_cust_count(int i): sets the count_cust as **i**.

Set_off_count(int i): sets the count_off as **i**.

Set_pres_count(int i): sets the count_pres as **i**.

Set_name(string n): sets the pharmacy name as **n**.

Get_name(): returns the pharmacy name.

Get_med(int i): returns the medication in **med** array with the index **i**.

Get_cust(int i): returns the customer in **cust** array with the index **i**.

Get_off(int i): returns the offtheshelf medication in **off** array with index **i**.

Get_pres(int i): returns the prescription medication in **pres** array with index **i**.

Set_cust(Customer o, int i): takes a Customer object and an index. Checks if the index is a valid index, then sets the customer at index **i** with the customer object **o**.

Set_pres(Prescription o, int i): takes a Prescription object and an index. Checks if the index is a valid index, then sets the prescription medication at index **i** with prescription object **o**.

Set_off(OffTheShelf o, int i): takes an OffTheShelf object and an index. Checks if the index is a valid index, then sets the offtheshelf medication at index **i** with offtheshelf object **o**.

Set_med(Medication o, int i): takes a Medication object and an index. Checks if the index is a valid index, then sets the medication at index **i** with medication object **o**.

num_of_med(): returns the number of medications in pharmacy.

Num_of_cust(): returns the number of customers in pharmacy.

Num_of_off(): returns the number of offtheshelf medications in pharmacy.

Num_of_pres(): returns the number of prescription medications in pharmacy.

Total_cash(): returns the total price of stock in pharmacy.

DisplayInfo(): prints the detailed information about the pharmacy.

Main File

resizePharma(Pharmacy* &pharma, int &prevsize): Asks the user for the max amount of pharmacies. Then resizes the **pharma** array based on the minimum between new size and previous size. Returns the new size

addPharmacy(Pharmacy* &pharma, int &pharmCount, int max): Checks if we have reached the max amount of pharmacies. If we did, it will ask the user if they want to resize the array of pharmacy. Then it will ask the user for the pharmacy name and add a pharmacy to the array of pharmacy.

addMedication(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then to enter the type of medication (1 for medication, 2 for off the shelf, 3 for prescription). And to enter the details of the medication depending on the type. Then adds the medication to the pharmacy.

removeMedication(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then to choose if they want to remove by name or index. Then it will remove the medication from the pharmacy based on what they enter next.

addCustomer(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then to enter the customer details to be added. Then adds the customer to the pharmacy.

printMed(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then prints the available medications in the pharmacy.

printCust(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then prints the customers in the pharmacy.

printPharmInfo(Pharmacy* &pharma, int pharmCount): Asks the user for pharmacy id. Then prints the pharmacy info.

Main(): The main function which has all the variables. It outputs the menu for the user to choose from. And based on the user choices it calls the correct function.