

Assignment 03 – Memory Management & Operator Overloading

Average expected estimated time to complete this assignment is 12 hours

- [1 HOUR] software requirement understanding
 - reading of the assignment document
 - understanding of problem statement
 - understanding of software inputs and outputs
- [2 HOURS] software design
 - identification of user-software interaction sequences
 - software menus
 - user inputs and
 - software flows
 - identification of required variables and data structures to be used
 - for input, processing, and output
 - identification of storage class specifications of required variables
 - modularization of software into required functions
 - proper naming of functions
 - identification of tasks to be performed into functions
 - identification of parameters and return types of functions
 - identification of user defined header files
 - names and functions to be placed in
 - flow sequence of main function and call of various user defined functions from it
- [6 HOURS] software coding
 - coding of user defined functions
 - placement of user defined functions into header file/s
 - coding of main function
- [2 HOURS] software testing
 - running software on various inputs
 - identification of software error and bugs
 - removal of software error and bugs
 - [1 HOUR] software documentation
 - proper indentation of code
 - use of proper naming conventions
 - commenting of code

Objectives:

- This assignment will provide experience with Class, Constructors, Destructor, Member Functions, Operator Overloading

Prerequisite Skills:

- Understanding of Class design
- Understanding of Constructor / Destructor
- Understanding of Operator Overloading
- Understanding Memory Management

Before you start:

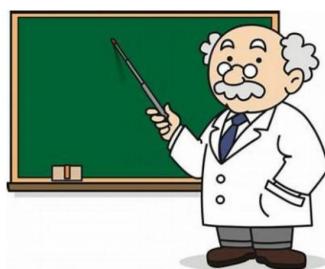
Please consider the following guidelines before you start the Assignment:

- Make sure that you understand the complete assignment before you actually start implementing the code.
- Your code must be in clean and easy to read form.
- Please handle all the corner cases and input validations of parameters inside functions. Functions without the exception handling wont be graded full marks.
- For any kind of understanding issues, please consult your TAs.
- After the assignment deadline, we will have a Viva for this assignment.
- ***Attention: Any form of cheating detected will result in 0 marks in your OOP sessional activities. So avoid cheating from each other and using AI tools.***

Pharmacy Management System

Hi! Class, meet **Mr. Clark**. Mr. Clark has heard that you guys are going incredible in OOP 😊. So he is here to have a little test of all of you. Over to Mr. Clark now. With him is Mr. Phil. While Mr. Clark is about to test you, Mr. Phil is here to guide you along the way. He often shares a word of advice whenever he feels need to. So keep a close ear (I mean eye) to Mr. Phil words. With him is Detective Oscar. He is very suspicious and often thinks out of the box about the corner cases in programs which can throw exceptions. He will let you know whenever he thinks an exception might occur here. Pay attention to him to make your program robust and error free.

Mr Clark: Hi! class, I am Prof. Clark. I am here to have a little OOP programming exercise with you. So lets get to business.



Students, My friend Mr. John opened a new pharmacy. He needs your help to track the medicines and all of his pharmacy customers and staff. With the knowledge of OOP I would like to give you a program which will help the John in achieving his goal.

Already made a Date Class So you will have to use that in this assignment

➤ ADDRESS CLASS

This class provides members and methods to help John keeping the address of his customers and staff.

Properties:

- City_name
- Email
- Street_name
- Mobile No [should be vector]

Mr Phil:



Mobile numbers should be unique. Handel this at class level.

Methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values
- **Destructor**

Detective Oscar:



Why not to put validation on email? Valid email domains are: gmail, outlook and hotmail . Also email address are unique

- **Getter and Setter for all Date Members**
- **Display method:** Display the members in good formatting form

Overload the following Operators:

- **[index] :** This operator ([]) will return the mobile no of given index.



Index must be greater than zero and less than size of the vector.

- **<< :** This will display only city name and email of the Address class.
- **>> :** This operator will only take input of city, street name and email .

➤ GENERAL_INFO CLASS

This class will hold the common information required for customers and staff.

Data member:

- *First_name*
- *Last_name*
- CNIC
- Address Object
- Gender [*We live in strange times, so gender will be an array.*]



*CNIC should be 13 digits all numbers.!!
CNIC Should be unique!!*

Methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values
- **Destructor**
- **Getter and Setter for all Date Members.**
- **Display method:** Display the members in good formatting form

Overload the following Operators:

- **Assignment operator**
- **<< :** This will display first name and last name.
- **>> :** This operator will only take input of first name, last name and cnic

➤ CUSTOMER CLASS

This class will hold information required about customer.

Data member:

- *Customer_ID [should be unique , sequentially increase]*
- *General_info object [To hold the information]*
- Membership_status [is a member customer of the pharmacy or not]

methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values

- **Destructor**
- **Getter and Setter for all Date Members**
- **Display method:** Display the members of the customer class

Overload the following Operators:

- **Assignment Operator**
- **<< :** This will display customer id and payment.method
- **>> :** This operator will only take input of payment method.

➤ **STAFF CLASS.**

This class will hold information required about staff.

Data members:

- **ID:** A unique identifier for each staff member, sequentially increased.
- **Role:** The role or position of the staff member in the pharmacy (e.g., pharmacist, technician, assistant).
- **Employment Date:** The date when the staff member started working at the pharmacy.
- **General_info object:** This will hold the other information.
- **salary**

methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values
- **Destructor**

- **Getter and Setter for all Date Members**
- **Display method:** Display the members of the customer class

Overload the following Operators:

- **Assignment Operator**
- **<< :** This will display id, role and employment date.
- **>> :** This operator will only take input of role and employment.

➤ **MEDICATION CLASS**

This class will hold information required about medicine.

Data member:

- **Medication ID:** Generated Sequentially .
- **Medication Name**
- **Description**
- **Price**
- **Quantity in Stock**
- **Expiry Date**
- **isExpired:** Return true false depending on the expiry date
- **Company object:** This holds the company information of which medicine belongs

methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values
- **Destructor**

- **Getter and Setter for all Date Members**
- **Display method:** Display the members of the customer class
- **isExpiredOrNot**

Overload the following Operators:

- **Assignment Operator**
- **<< :** This will display id, name, price Quantity Stock, expiry date.
- **>> :** This operator will only take name, price, quantity stock and expiry date.

➤ **Pharma COMPANY CLASS**

This class will hold information required about medicine.

Data member:

- Company ID: Generated uniquely, Sequentially increased.
- Company Name
- Location
- List of medicine they sell

Methods:

- **Default constructor:** This constructor will initialize the data member with default values.
- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
- **Parameterized construct** This will initialize the data members with given values
- **Destructor**

- **Getter and Setter for all Date Members**
- **Display method:** Display the members of the customer class

Overload the following Operators:

- **Assignment Operator**
- << : This will display id, company name and location.
- >>: This operator will only take input of company name and location.

➤ SALESMAN CLASS

This class is to represent a salesman. A pharmacy can only buy medicines from a salesman. A salesman will be from a Pharma Company.

Data members:

- Salesman Id (uniquely assigned automatically).
- General info object.
- Pharma Company (object).
- Set_commission (This is the agreed commission company will pay the salesman, for each medicine he will sell.)
- Obtained_commission (The total obtained commission by salesman, company pays 5% of each medicine sold by the salesman.).

Pharmacy owners will interact with salesmen. Salesmen will show them their company medicines. The Pharmacy owners will select from those medicines and purchase.

Methods:

- **Regular methods**
- Method to show all the medicines of the Company. User should be able to select multiple medicines at a time by providing id numbers and those medicines should be added to his pharmacy.
- **Getters and setters.**

- Display function.
- Add commission

➤ **ORDER CLASS**

This class is to hold information about an order made by a customer.

Data members:

- Order id (unique ,self generated)
- Customer id
- Order date (Date object)
- Bought medicine [D- array of medicine]
- Salesman
- Company
- Total amount
- Payment method

Methods:

Provide the necessary functions.

➤ **PHARMACY CLASS**

This class provides admin to manage medicines and customers

Data member:

- Pharmacy ID: Unique and sequentially generated
- Pharmacy Name
- Pharmacy owner.
- Dynamic Array of medication
- Max Size and Current Size of medication
- D-Array of quantity of each medicine
- Dynamic Array of Staff
- Max Size And Current Size of Staff
- Dynamic Array of Salesmen
- Max Size And Current Size of Salesmen
- Dynamic Array of Orders
- Max Size And Current Size of orders array
- Total sales
- Total purchases
- Total profit

Methods:

- **Copy Constructor :** There will be Copy constructor which will copy all the elements of other object using Deep copy.
 - **Parameterized constructor** This will initialize the data members with given values. Size can be given for all dynamic array.
 - **Destructor**
 - **Add a new Medication** [Resize the array if needed] (new medication will be added via a salesman.)
 - Remove a Medication by its ID [Resize the array if needed]
 - Add a new Order [Resize the array if needed]
 - Add a new Staff [Resize the array if needed]
 - Remove a Staff by its ID [Resize the array if needed]
 - Add a new Salesman [Resize the array if needed]
 - Remove a Salesman by its ID [Resize the array if needed]
 - Display method to show all orders
 - Display method to show all staffs
 - Display method to show all medication
 - TotalPurchase :[display total purchase]
 - TotalSales [It will return the total sales]
 - Display Method [to display everything]
 - Update medicine details by its ID
 - Sell Medicine [it will take input for customer and then shows the list of medicine and customer will select all required medicine. Make the order object and, do necessary modifications in your pharmacy and place that order in order array.]
 - Purchase medicine: purchase medicine from a salesman. Make necessary changes as well.
 - Show all the orders made.
 - Show all the orders made by a particular customer.
 - Show all the orders made for a particular medicine.
 - Show all the orders made on a specific date.
 - Show all the Salesmen.
 - Method to pay the salary of all the staff. Amount will be deducted from profit.
 - DiscardExpiryMedicine [This method will remove the expiry medicine from array]
- Overload the following Operators:
- Assignment Operator
 - << : This will display everything about pharmacy.
 - >> : This operator will only take pharmacy name and owner name

Main Function:

Main function will have a D-array of Pharmacies. Make an independent function to add a new Pharmacy. Your Main function should have a menu to provide the user functionality to add a Pharmacy and also all the functions of pharmacy class. For each function, ask the user for which pharmacy they want to call that function.

.....

Corner cases: Being good Programmers, It is your responsibility to handle corner cases in your functions. Consider your user dumb, so program according to it. Your program is as good as your corner cases handling. (Chief! Handling corner cases will get you extra marks)

Congratulations! You have reached the end of your assignment. Well done. Keep up the good work.

