Object Oriented Programming (Spring-2023)

Assignment 4

Submission Deadline: 6th May, 2023 at 12:30 PM.

Instructions:

1. The usage of string library is allowed.
2. Your code must be **generic, use dynamically created arrays only.**
3. Correct and timely submission of the assignment is the responsibility of every student.
4. **Evaluation:** The assignment is of 200 marks. All submissions will be evaluated based on demo. Failure to full fill the requirements of question will result in loss of marks.
5. **Plagiarism:** Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. If found plagiarized, you WILL be awarded zero marks in the assignment.
6. Repeating such an act can lead to strict disciplinary actions and failure in the course.

NOTE: Please start early otherwise you will struggle with the assignment.

7. **Submission Guidelines:** Dear students, we will be using auto-grading tools, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.
   A. Make a folder titled **21X-XXXX_A2_OOP** and put all your .cpp files and .h files in it. Compress the folder as a zip file and upload on **google form** only.
   B. Submission: Create Header .h files for creating your classes and have your function definitions in .cpp file.
   C. You need to submit both header file and cpp file for each question.
   D. All cpp files must contain **your name, student-id, and assignment # on the top of the file** in the comments. Place all your .cpp files (only) in a folder named your ROLLNUM_SECTION (e.g. 21i-0001_A).
      i. You will zip the folder and submit it on **google forms.**
      ii. No other method of submission will be accepted.
      iii. Start the submission process well before time so that you can overcome problems as you face them.

Note: Follow the given instructions to the letter, failing to do so may result in a zero.

The goal of this assignment is to develop a program that implies OOP concepts that you have learned in your class.

Q1: Shopping Cart

(70 Marks)

Your goal here is to write classes for creating a shopping cart of items. To create the cart of items you will need to write the following two classes.
Design a class Item. A "Item" is characterized by the following **PRIVATE** attributes:

- Name
- ItemType
- Basic price per unit

The class has the following behaviors:

- A default constructor.
- Setter and getter functions

The following classes will be created next. Egg, Bread *{item type: Food}*, Pens *{item type: supplies}* and socks *{item type: clothes}.* These classes will have the following **PRIVATE** attributes.

- Number of Units purchased.
- an indication whether the item is on sale or not.
- And an indication to know whether the item comes with complementary gift card or not.

The class has the following behaviors:

- a constructor initializing the attributes using parameters given in the order shown by the provided main(); a default constructor will not be necessary but the last three parameters will have 1, false and false as default value. The name parameter won't be received but will be set to respective names in their respective constructors.
- a price() method returning the item's price: the price will be the base price if the item is not on sale; otherwise, the price will be 25% less than the base price.
- a bool gift() method indicating whether the item offers gift card or not.
- Overloaded stream insertion operator. The characteristics have to be displayed in strict accordance with the following format:
  <Name> <Unit Price> <No of Units> <on Sale> <Complementary Gift card>, Price: <Price> Rs.

- an overloading of the == operator returning true if two items are identical, false otherwise. Two items are considered identical if they have the same name, unit price, and the two items both belong to same item type or not. (the fact that the item is on sale or not is involved in the comparison).

Next, write a "Cart" class which will have the following attributes:

- Item *
- Static int total_items

The Cart class offers the following methods:

- Total_items will keep count of all the items of all carts.

- A method getTotalItems() that returns the number of items in all carts.
- a method bool Discount() returning true if the total cart is getting a discount and false otherwise; a cart is receiving a discount if at least three of its items are on sale.
- a method price() without parameters returning the price of the cart of items; This is the sum of the prices of all its items; this sum is reduced by 12.5% if the cart is getting a discount;
- operator string() const to return value of the Cart as a string as:
  If the cart does not contain any item,

  Still no item in the cart
  Else:
  <Item1>
  ..
  < ItemN>
  Total Price: <Price_of_cart> Rs.

- a stream insertion method, should display all information of cart with the total price. This method will display the characteristics of the items in cart respecting rigorously the following format:
  If the bouquet does not contain any flower,
  Still no item in the cart
  Else:
  Cart Contains

  <Item1>
  ..
  < ItemX>
  Cart is Discounted: Yes/No
  Total Price: <Price_of_Cart> Rs.

Here < ItemX > means display of the Xth item of the cart in the format specified by the overload of the << operator. There is a newline after displaying each item and after displaying the price of the cart.

- an overload of the += operator which allows adding a item to the cart, the item will always be added at the end.
- an overload of the -= operator taking as a parameter a item and removing from the cart all units of items belonging to item type of latter
- an overloaded + operator according its usage in the provided main
- an overloaded - operator according to its usage in the provided main

NOTE: Make sure no Memory is Leaked in your code

```
int main() {
	// example of non sale item.
	Bread p1(12, 3);
```

```
    cout << r1 << endl;

    // example of sale item
    Eggs p2(15, 1, true);
    // example of sale item with gift card
    Sock p3(20, 2.0, true, true);

    Cart c1;
    c1 += p1; //add bread to cart
    c1 += p1;
    c1 += p2; //add eggs to cart
    c1 += p3; //add socks to cart
    cout << c1 << endl;

    c1 = c1 - p1; // Delete all the unit of type bread
    cout << c1 << endl;

    Cart c2;
    c2 = c1 + p2; // Add item of type eggs
    cout << c2 << endl;

    // Delete all the units of eggs type food from the cart.
    c2 -= p2;
    cout << c2;
    cout << Cart :: total_items; //displays number of all items in all carts.
    return 0;
}
```

Q2: Maintenance Mania

(40 Marks)

A company looks after many small-scale maintenance projects. A project starts executing as soon as an employee is assigned to it. A project may or may not be supervised by a manager who will monitor the employee's performance. If an employee is not working on any project, he/she will be unsupervised until assigned to a new project.

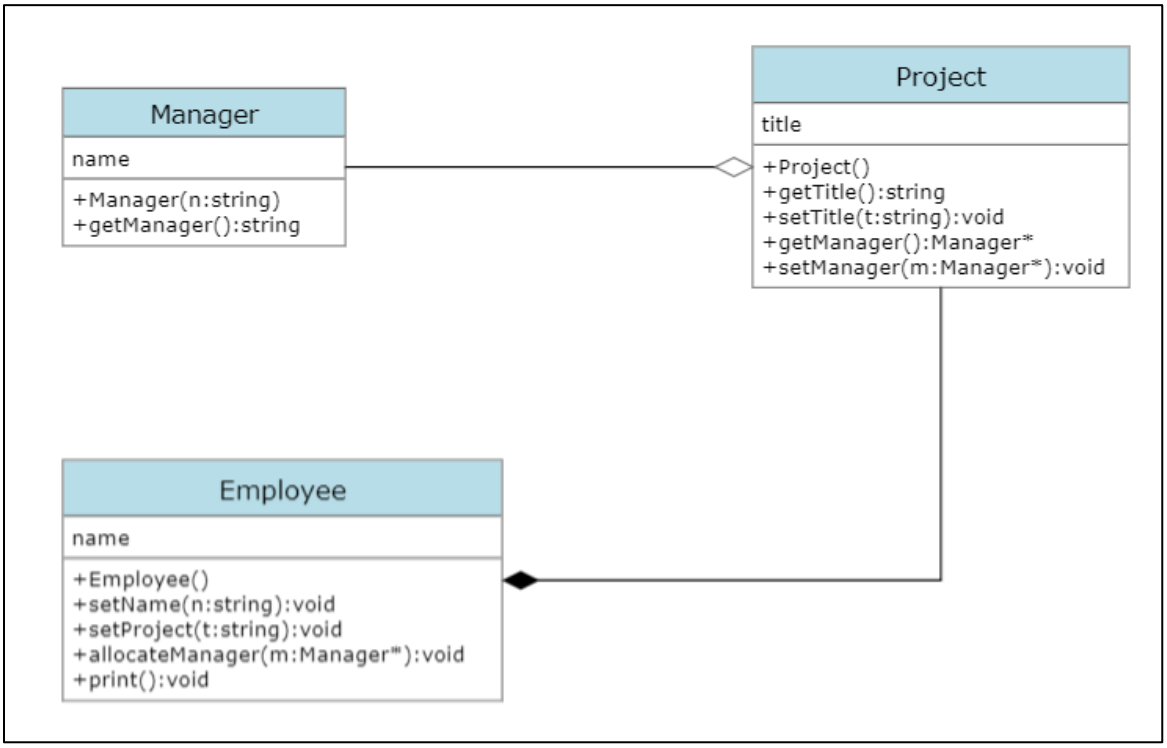
Write a complete C++ program based on the following classes:

1. Class **Manager**
    a. The constructor
    b. getManager()
2. Class **Project** has the following methods:
    a. the constructor
    b. getTitle()
    c. setTitle()
    d. getManager()

   e. setManager()
3. Class **Employee** has the following methods:
   a. the constructor
   b. setName
   c. setProject
   d. allocateManager
   e. print: to display the Employee's name and project's title, and also the Manager's name (but only if the Employee has a Manager).

4. Create four objects of Manager for "Michael", "Dwight", "Jim" and "Andy".
5. Then create an array of objects of Employee with maximum number of employees to store into the array is 15. There may or may not be a project for each employee.
6. Populate the array.
7. Assign managers to employees as follows: **a**. The first manager is assigned to be the supervisor for the first and second employee. **b**. The second manager is assigned to the third and seventh employee. The third manager will manage employees four through nine except seven. And the fourth manager will supervise project fourteen and fifteen.
8. Print all the employees.

NOTE: Make sure no Memory is Leaked in your code



Q3: Library Management System

(70 + 20 Marks)

In this question, you need to write a host of classes, and place them in a reasonable hierarchy as per given case study:

A librarian will be able to login into the system and can add new books to the library. They will also be able to search for a book in records, issue and receive a book back from the registered member of the library. and can generate chalan for fine in case of delay in return. The books in the library will be recorded by ID, name, author, edition number purchase date and marked available or issued in the system. Library will have magazines, journals, and Study books available for issuance. Any student or faculty member will be able to take books of any kind out of the library as long as they have a membership to library. To have membership, each person will have to have a registered ID, name, address and phone number with library admin. They will also have to identify themselves as students or Faculty as the issuance period will be different for both. Each person will be issued a card which will have an issuance limit. Students won't be able to issue more than 3 books at a time where a faculty will be able to issue up to 5 books. But they can always issue less than authorized quantity. The same book won't be issued twice at the same time to two different people.

Identify classes, attributes, and functions for above scenario. **Implement the case study using C++ concepts of inheritance, composition, and aggregation where and if necessary.**

NOTE: **you will also create a class diagram of case study and submit its pdf along with C++ implementation. (20 marks)**

NOTE: Make sure no Memory is Leaked in your code

You need to make a MENU driven interface and make the following scenario :

1. Create a new Librarian.
2. Librarian should be able to Login and add 10 books to the library database.
3. Create 3 students and 2 Faculty members.
4. Student 1 should be able to issue 2 studyBooks and a magazine, student 2 should be able to issue 1 journal. Student 3 can issue a magazine and a study book. Same book can't be issued twice.
5. Staff member should be able to issue 2 journals and a studybook.
6. Due date for Students should be 14 days and the deadline for staff members should be 2 months.
7. Display return date of all the books issued by Student 1.
8. Return the books issued by student 2. Display fine for the book returned late.
9. Update the status of returned book & user.

---------- **Happy EID** 😊 ----------