CL-1004
Object Oriented Programming
Lab No 9

Objectives:

- Classes and objects
- Data members
- Member function
- Data encapsulation
- Member access specifier (private , public , protected)
- Constructor, Destructor
- Copy Constructor
- This pointer
- Constant data members
- Inheritance
- Single Level inheritance
- Multi-level inheritance
- Multiple inheritance
- Diamond Problem

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function and about its functionality.
3. Mention comments where necessary such as comments with variables, loop, classes etc to increase code understandability.
4. Use understandable name of variables.
5. Proper indentation of code is essential.
6. Write a code in C++ language.
7. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every task **output in Microsoft Word and submit word file. Do not submit .cpp file.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google Classroom. (Make sure your submission is completed).
11. Please submit your file in this format **20F1234_L1**.
12. Do not submit your assignment after deadline. Late submission is not accepted.
13. Do not copy code from any source otherwise you will be penalized with negative marks.

Problem 1: | (Inheritance)

Design a class named **PersonData** with the following member variables:

- FirstName
- LastName
- Address

Write the appropriate **getter** and **setter** functions for these member variables.

Next, design a class named **CustomerData**, which is derived from the **PersonData** class. The CustomerData

Class should have the following member variables:

- CustomerNumber
- MailingList

The CustomerNumber variable will be used to hold a unique integer for each customer. The MailingList variable should be a bool. It will be set to true if the customer wishes to be on a mailing list, or false if the customer does not wish to be on a mailing list. Write appropriate getter and setter functions for these member variables. CustomerData class will have the

- InputCustomerData member function which will Input all the data for customer.
- DisplayCustomerData member function which will display all the data for customer.

Demonstrate an object of the CustomerData class in a simple program.

Use three file structure

Problem 2: | (Multi level Inheritance)

A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer's discount is determined by the amount of the customer's cumulative purchases in the store.

- When a preferred customer spends $500, he or she gets a 5% discount on all future purchases.
- When a preferred customer spends $1,000, he or she gets a 6% discount on all future

purchases.

- When a preferred customer spends $1,500, he or she gets a 7% discount on all future purchases.
- When a preferred customer spends $2,000 or more, he or she gets a 10% discount on all future purchases.

Design a class named PreferredCustomer, which is derived from the CustomerData class you created in **problem 1**. The PreferredCustomer class should have the following member variables:

- purchasesAmount (a double)
- discountLevel (a double)

The purchasesAmount variable holds the total of a customer's purchases to date. The discountLevel variable should be set to the correct discount percentage, according to the store's preferred customer plan. Write appropriate member functions for this class and demonstrate it in a simple program.
Use three file structure

Problem 3: | (Multi Level Inheritance)

Write a class **Person**. It should have the following three attributes:

A. Name
B. Age
C. Gender

Now do the following steps

- Inherit a class **EmployedPerson** from person.
- Employed should inherit only Name from the Person.
- Employed should have an attribute NIC number.
- Employed has a member function Employ that should output "Hi, I am Employ from Employed Class".

- Inherit another class **Unemployed** from Person.
- Unemployed should inherit Name from Person.
- Unemployed should have no other attribute but constructor should output "Hi, I am UnEmploy from UnEmployed Class".

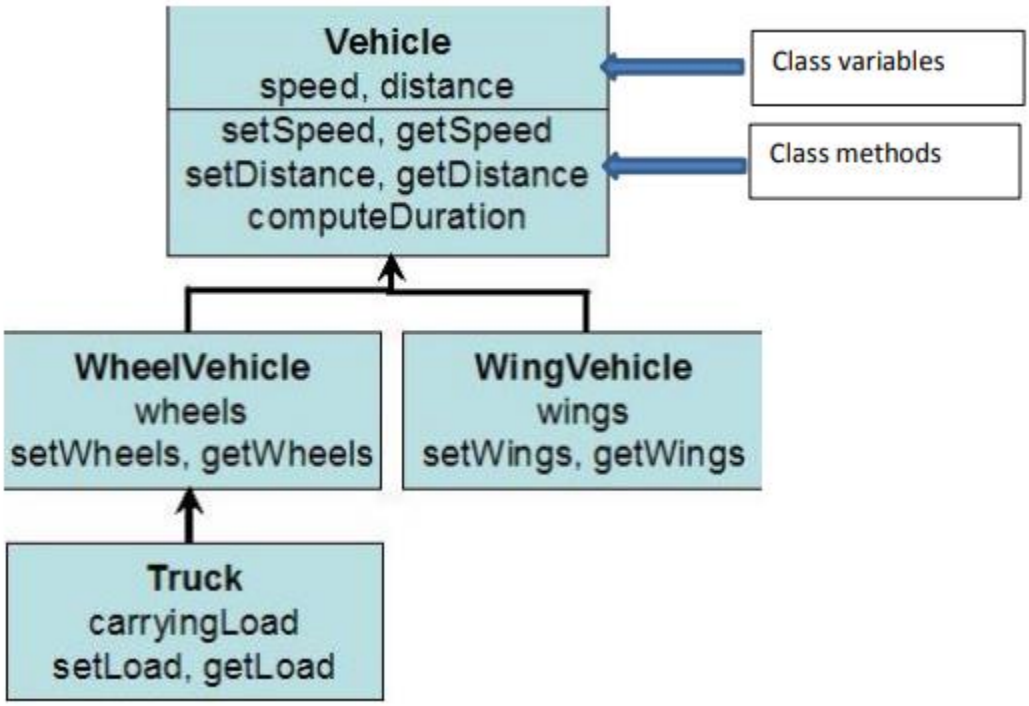
- Inherit a class **BusinessMan** from Employed.
- It should inherit only NIC number from Employed.
- BusinessMan should inherit Employ function from Employed.
- Write a display function in this class. It should display all the attributes of BusinessMan in it.

- Create an object of BusinessMan in main and call the Employ function.

Call the display function of BusinessMan with the same object.
Create an object of UnEmployed in main and print the members. Note: Only write default constructors for each class. Default constructor should initialize all attributes of class to appropriate values.
Use three file structure

Problem 4: | (Multi Level Inheritance)

Design the following classes with given variables and methods using inheritance.



Explain your work with proper output.
Use three file structure.

Problem 5: | (Multiple Inheritance)

Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. The constructors of inherited classes are called in the same order in which they are inherited.

For example, in the following program, B's constructor is called before A's constructor
Write the main for following classes

```cpp
#include<iostream>
using namespace std;

class A
{
public:
A() { cout << "A's constructor called" << endl; }
};

class B
{
public:
B() { cout << "B's constructor called" << endl; }
};

class C: public B, public A // Note the order
{
public:
C() { cout << "C's constructor called" << endl; }
};
```

The output for the main should be like this

```
B's constructor called
A's constructor called
C's constructor called
```

Use three file structure

Proper code indentation will hold extra marks!

Best of luck ☺

You are done with your exercise, submit on classroom at given time.