CL-1004
Object Oriented Programming
Lab No 10

Objectives:

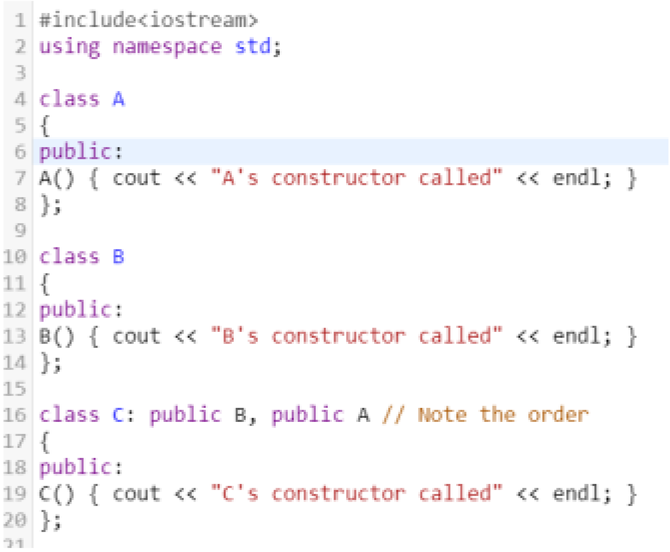
- Classes and objects
- Data members
- Member function
- Data encapsulation
- Member access specifier (private , public , protected)
- Constructor, Destructor
- Copy Constructor
- This pointer
- Constant data members
- Inheritance
- Single Level inheritance
- Multi-level inheritance
- Multiple inheritance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

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: | (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



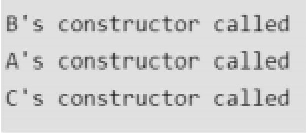
```cpp
#include<iostream>
using namespace std;

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

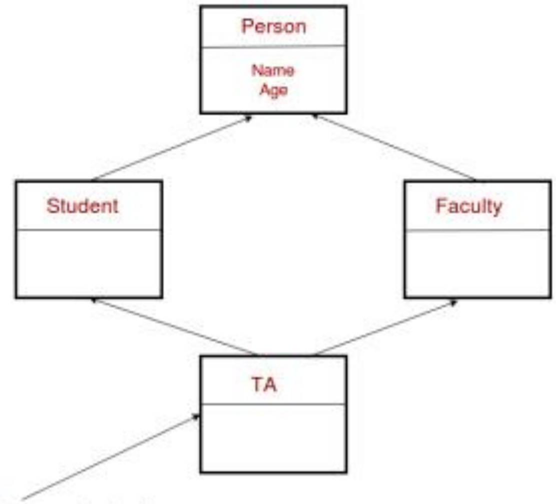


```
B's constructor called
A's constructor called
C's constructor called
```

Use three file structure

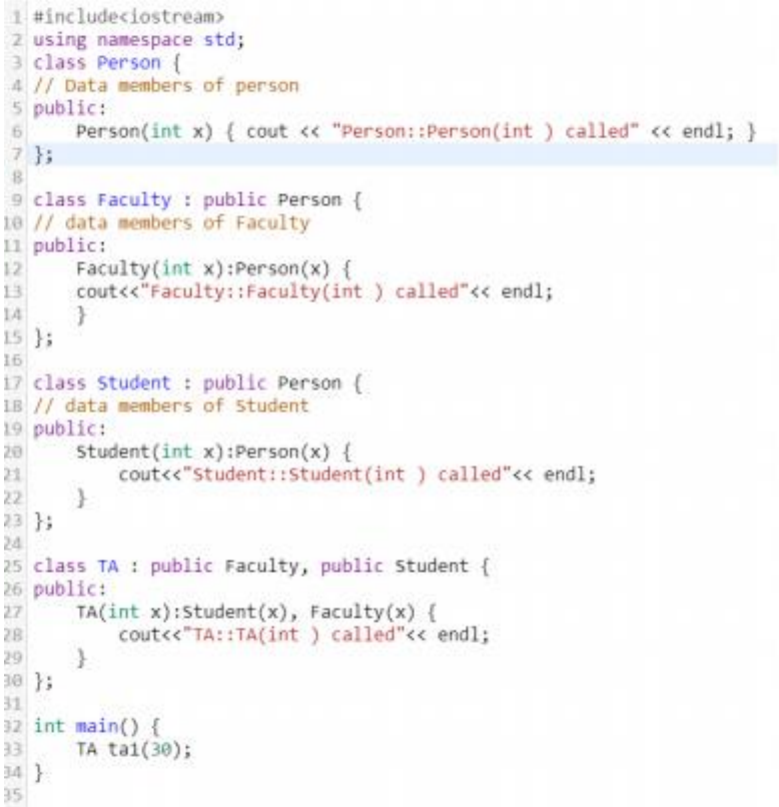
Problem 2: | (Multiple and Multi Level Inheritance, Diamond Problem)

The diamond problem occurs when two super classes of a class have a common base class. For example, in the following diagram, the TA class gets two copies of all attributes of Person class, this causes ambiguities.



Name and Age needed only once

a) What is the output for below code?



```
#include<iostream>
using namespace std;
class Person {
// Data members of person
public:
    Person(int x) { cout << "Person::Person(int ) called" << endl; }
};

class Faculty : public Person {
// data members of Faculty
public:
    Faculty(int x):Person(x) {
    cout<<"Faculty::Faculty(int ) called"<< endl;
    }
};

class Student : public Person {
// data members of Student
public:
    Student(int x):Person(x) {
        cout<<"Student::Student(int ) called"<< endl;
    }
};

class TA : public Faculty, public Student {
public:
    TA(int x):Student(x), Faculty(x) {
        cout<<"TA::TA(int ) called"<< endl;
    }
};

int main() {
    TA ta1(30);
}
```

b) Mention problem if any in **part a** and implement it correctly.

c) Write the code in 3-file structure.

Problem 3: | (Multi Level Inheritance, Function Overriding)

Write a program which can detect the sounds of Animals.
Write a class Animals having the following attributes:

1. String Name

2. Sound()

Inherit the following classes from Animals having the same attributes and behavior of base class
Override the sound function for each class

1. Cat
2. Dog
3. Tiger_Family
4. Deer

From Tiger_family class inherit the following sub classes publically: (override the sound function for each class).

1. Tiger
2. Lion

Note:

- Make instance of each class and call sound function.
- Each class animal should have its own sound.
- Program should be in 3-file structure.

Problem 4: (Polymorphism, Virtual function, abstract and concrete classes)

Design a **Ship** class that has the following members:

- A member variable for the name of the ship (a string)
- A member variable for the year that the ship was built (a string)
- A constructor and appropriate **accessors** and **mutators**
- A **virtual print** function that displays the ship’s name and the year it was built.

Design a **CruiseShip** class that is derived from the Ship class. The **CruiseShip** class should have the following members:

- A member variable for the maximum number of passengers (an int)
- A constructor and appropriate **accessors** and **mutators**
- A print function that overrides the print function in the base class.

The **CruiseShip** class’s print function should display only the ship’s name and the maximum number of passengers. Design a **CargoShip** class that is derived from the Ship class. The **CargoShip** class should have the following members:

- A member variable for the cargo capacity in tonnage (an int).
- A constructor and appropriate **accessors** and **mutators**.
- A print function that overrides the print function in the base class.

The **CargoShip** class’s print function should display only the ship’s name and the ship’s cargo capacity. Demonstrate the classes in a program that has an array of Ship pointers. The array elements should be initialized with the addresses of dynamically allocated Ship, The program should then step through the array, calling each object’s print function.

Problem 5: (Polymorphism, Pure Virtual function, abstract and concrete classes)

Finding the areas of Polygon family (Square, Rectangle and Triangle) using Polymorphism and Pure Virtual function.

Define a base class Polygon having the following attributes:

1. Float length
2. Float width

Pure Virtual Member functions:

1. Float Area() to calculate Area
2. Float Perimeter() to calculate Perimeter
3. Void Display() to display the calculated Area and Perimeter

Inherit the following classes from Polygon publically (Inherit all data members and member functions):

1. Square
2. Rectangle
3. Triangle

Task:

1. Write default constructor and Overloaded constructor to overload the values of Length and Width.
2. Calculate the Areas and Perimeters of Polygon family.
3. Show the results on the console using Display () function.
4. Write base class pointer "*Bptr" which points to the objects of derived classes one by one and display the area and perimeter of subclasses using Dynamic Binding

Hints:

- Square Area/Perimeter=4*L
- Rectangle A=2*(LxW) P=2*(L+W)
- Triangle A= (LxW)/2 P= (L+W)

Problem 6: (Polymorphism, abstract and concrete classes)

Design a software system for FESCO-WAPDA for 2020 from which they calculate the monthly bill of Electricity.

Write a class Bill having the following attributes and behavior

1. Int Units
2. Int per_unit_cost
3. Float calBill
4. Monthly_bill()
5. Bill_display()

Derived the following classes from the base class Bill:

1. January_bill
2. February_bill
3. March_bill
4. April_bill
5. May_bill

Inherit the attributes and behavior of base class. Perform the following tasks.

Task:

Calculate the monthly bills for every month.

1. Display the total bill using function Bill_display() method.
2. From main() use base class pointer to point to derived classes objects to calculate the total bill for each month.
3. Call destructor when you realize that the objects are no more needed.
4. Use virtual destructor concept in the above hierarchy.

Note:

Use virtual Member functions for derived classes.

- Use appropriate values for per unit cost of monthly bills.
- User appropriate data types.

Proper code indentation will hold extra marks!

Best of luck ☺

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