**Concept to be implemented**: friend function and friend class

**Lab Assignment 6**

|  |  |
| --- | --- |
| **Roll No.: A055** | **Name: Ibrahim Shaikh** |
| **Program: B. Tech**-**CSBS (2ND YEAR)** | **Date of Release: 30th August 2021** |
| **Batch:** 1/A | **Date of Submission: 11th August 2021** |

**PROBLEM STATEMENT:**

**Problem Statement 1:** Examination department of NMIMS wants to develop an application to calculate the results depending on ICA and TEE marks. They accept marks of ICA and TEE from different sources and combine them as internal and external marks out of 100. Therefore, two separate classes as “ICA” and “TEE” with following data members are created. (out of 50 marks for each class)

* + - * Marks of Math
      * Marks of OPP
      * Marks of SE
      * Marks of DS

Use member functions to input the marks for each class. Identify a concept and write a function that can accept data members of both the classes and calculate the results. Same function will print the result and total marks subject wise.

**Problem Statement 2:** Write a program to demonstrate friend class behaviour. Friend class functions will access the data members of the other class.

1. Write a class “Employee” to read and print information of an employee with following details:

* Name of the employee,
* Id of the employee,
* Department of the employee
* monthly salary

1. Write another class “Sales” that read and print employee details as follows

* Sales in month,
* Incentives acquired

Calculation of Incentives will be as follows:

1. 10 % of the monthly salary if sales is between 50 to 100 units
2. 20 % of monthly salary if sales is between 100 and 150 units
3. 30 % of monthly salary if sales is more than 150 units.

Later on company has decided to include performance indicator depending on total sales made in a month as Excellent, Good, Satisfactory or Poor. Add one more function in sales class to assign performance indicators to the employee.

Use following conditions:

* Sales above 150 – Excellent
* Sales between 101 to 150 – Good
* Sales between 50 to 100 – Satisfactory
* Sales below 50 - Poor

Write main () function to implement the friend class and perform the operation.

**CODE:**

**Code of Problem Statement 1**

#include<iostream>

using namespace std;

class TEE;

class ICA

{

    float m, o, s, d;

    public:

    void sdata()

    {

        cout<<"Enter the ICA Marks of Math (out of 50): ";

        cin>>m;

        cout<<"Enter the ICA Marks of OOP (out of 50): ";

        cin>>o;

        cout<<"Enter the ICA Marks of SE (out of 50): ";

        cin>>s;

        cout<<"Enter the ICA Marks of DSA (out of 50): ";

        cin>>d;

    }

    friend void add(ICA, TEE);      // friend function declaration

};

class TEE

{

    float ma, oo, se, ds;

    public:

    void readdata()

    {

        cout<<"\nEnter the TEE Marks of Math (out of 50): ";

        cin>>ma;

        cout<<"Enter the TEE Marks of OOP (out of 50): ";

        cin>>oo;

        cout<<"Enter the TEE Marks of SE (out of 50): ";

        cin>>se;

        cout<<"Enter the TEE Marks of DSA (out of 50): ";

        cin>>ds;

    }

    friend void add(ICA, TEE);      // friend function declaration

};

void add(ICA i, TEE t)

{

    cout<<"\nTotal Marks of Math (out of 100): "<<i.m+t.ma;

    cout<<"\nTotal Marks of OOP (out of 100): "<<i.o+t.oo;

    cout<<"\nTotal Marks of SE (out of 100): "<<i.s+t.se;

    cout<<"\nTotal Marks of DSA (out of 100): "<<i.d+t.ds;

}

int main()

{

    ICA a;

    TEE b;

    cout<<"Internal Marks:\n";

    a.sdata();

    cout<<"\nExternal Marks:";

    b.readdata();

    cout<<"\nTotal Marks:";

    add(a, b);

    return 0;

}

**Code of Problem Statement 2**

#include<iostream>

#include<string.h>          //for using 'string'

using namespace std;

class Employee             //Class Declaration

{

    string name, dept;

    int id, bsal;

    float da, hra, ta, gsal;        //Private data members

    friend class Sales; //Friend class declared

    public:                         //Public member functions

    Employee()          //Constructor

    {

        name = "NA";

        dept = "NA";

        id = -1;

        bsal = -1;

    }

    Employee(string a, string d, int i, int sal) //Constructor with Parameters

    {

        name= a;

        dept= d;

        id= i;

        bsal = sal;

        da = 0.5\*sal;

        hra = 0.3\*sal;

        ta = 0.1\*sal;

        gsal = bsal+da+hra+ta;

        return;

    }

    void display(int n)         //Display function to show output

    {

        int i = n;

        cout<<"\n\nName of the Employee "<<i<<": "<<name;

        cout<<"\nID of the Employee "<<i<<": "<<id;

        cout<<"\nDepartment of the Employee "<<i<<": "<<dept;

        cout<<"\nBasic Salary of the Employee "<<i<<": "<<bsal;

        cout<<"\nDearness Allowance of Employee "<<i<<": "<<da;

        cout<<"\nHRA of Employee "<<i<<": "<<hra;

        cout<<"\nTA of Employee "<<i<<": "<<ta;

        cout<<"\nGross Salary of Employee "<<i<<": "<<gsal;

    }

};

class Sales //friend class

{

    int smon, iacq;

    public:

    Sales()

    {

        smon = -1;

    }

    Sales(int sales)

    {

        smon = sales;

    }

    void print(int n, Employee a) //Employee a to call objects from //class Employee

    {

        int i=n;

        cout<<"\nSales of Employee "<<i<<" in this month: "<<smon;

        if(smon>=150)

        {

            cout<<"\nIncentives acquired by Employee "<<i<<": "<<a.bsal\*0.3;

            cout<<"\nEmployee Perfomance: Excellent";

        }

        else if(smon>=101 && smon<150)

        {

            cout<<"\nIncentives acquired by Employee "<<i<<": "<<a.bsal\*0.2;

            cout<<"\nEmployee Perfomance: Good";

        }

        else if(smon>=50 && smon<100)

        {

            cout<<"\nIncentives acquired by Employee "<<i<<": "<<a.bsal\*0.1;

            cout<<"\nEmployee Perfomance: Satisfactory";

        }

        else

        cout<<"\nEmployee Perfomance: Poor";

    }

};

int main()

{

    int n, i, sal, sales;

    string a, d;

    Employee e[10];            // Declaration of array of objects

    Sales s[10];

    cout<<"Enter the number of Employees (max. 10): ";

    cin>>n;

    if(n<11)

    {

    for(int x=1; x<=n; x++)     //for loop for taking user input

    {

        cout<<"\nEnter the name of Employee "<<x<<": ";

        cin>>a;

        cout<<"Enter the ID of Employee "<<x<<": ";

        cin>>i;

        cout<<"Enter the Department name to which Employee "<<x<<" belongs: ";

        cin>>d;

        cout<<"Enter the Basic Salary of Employee "<<x<<": ";

        cin>>sal;

        cout<<"Enter the Sales of Employee "<<x<<": ";

        cin>>sales;

        e[x] =Employee(a, d, i, sal);

        s[x] =Sales(sales);

    }

    for(int x=1; x<=n; x++)

    {

        e[x].display(x);

        s[x].print(x, e[x]);

    }

    }

    else

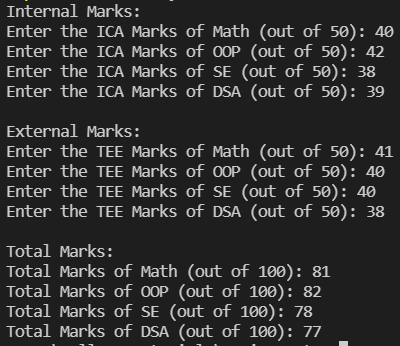
    cout<<"EXCEEDED MAXIMUM NUMBER OF EMPLOYEE!!!";

    return 0;

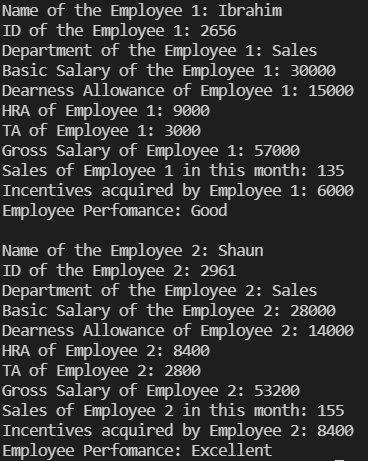
}

**OUTPUT:** 

**Output for Problem Statement 1**

****

**Output for Problem Statement 2**



**LINK FOR THE CODE:** Done using VSCode