

## O O C

## Assignment No: 1

**Problem Statement:** Develop an ~~obj~~ object oriented program in C++ to create a database of student information systems containing the following information. Name, Roll number, Class, division, Date of birth, Aadhar no., Blood group, Contact address, telephone no, etc. Construct the database with suitable member functions for initializing and destroying the data. Count the number of student in database by using a variable count which is a member of a student class. Also display the student information using a static function. Display the % marks obtained using friend function.

**Objectives:**

- 1) To learn concepts of classes and objects in C++.
- 2) To learn static members and friend functions in C++.

**Theory:**

Explain

## 1) Class.

→ A class is a blue print for creating objects.

→ It encapsulates data &amp; functions that operate on that data.

→ Class promote code reusability and modularity.

→ Eg.

Class car &amp;

public:



```

    string brand;
    int year;
    void display() {
        cout << "Brand: " << brand;
    }
};

```

## 2) Object :

→ An object is an instance of class. It represents a real world entity with attributes and behaviours defined by the class.

eg.

Car myCar;

myCar.brand = "Kia";

myCar.year = 2022;

myCar.display();

## 3) Static members:

→ Static members belong to the class rather than any object instance.

→ They are shared among all objects of the class.

→ A static member function can only access static data members.

Eg.

class Example {

public:

static int count;

static void incrementCount()

{ count++;

}

};



#### 4) Friend Function.

- A friend function is a function that is not a member of a class but has access to its private & protected members.
- It is declared by using the 'friend' keyword inside the class.

#### Algorithm :

1. Start
2. Create a student class
3. Create a function to accept student information
4. Create a function to display student information
5. Count the number of students in the database using a variable count which is a member of the student class.
6. Also display the student information using a static function.
7. Display the % marks obtained using friend function.
8. Stop.

Platform: 64-bit open source linux

#### Input:

Student Information: Name, Roll Number, Class, division, Date of Birth, Adhar number, Blood group, Contact address, telephone number.

#### Output:

Student Database containing Name, Roll number, Class, division, Date of Birth, Adhar number, Blood group, Contact address, telephone number.



Conclusion:

Hence, learnt to create a class, use static members and create friend function in C++.

FAQ's:

1) Differentiate between classes and objects.

	Classes	Objects
1)	Classes are blueprints used to create objects.	Objects are instances of classes.
2)	Memory is not allocated for data members.	Memory is allocated for data members.
3)	Define properties and behaviours.	Represents a real world entity.
4)	Used to declare and define methods of variables.	Used to access and manipulate class methods & variables.

2) Explain the concept of array of objects.

→ An array of objects is a collection of objects of the same class, stored in continuous memory locations.



- This allows for managing multiple objects efficiently, using array indexing to access individual objects.
- Using arrays of objects simplifies handling multiple instances of a class, making it easier to organize and process data in bulk.

3) What are limitations of static members?

→ The limitations of static members are as follows:

- Shared across all instances of the class.
- Cannot directly access non-static members.
- Limited to the class scope.
- Cannot be initialized inline within the class.
- Persist for the program's lifetime, potentially increasing the memory usage.
- Static member functions do not support polymorphism.
- Static member functions cannot be virtual.

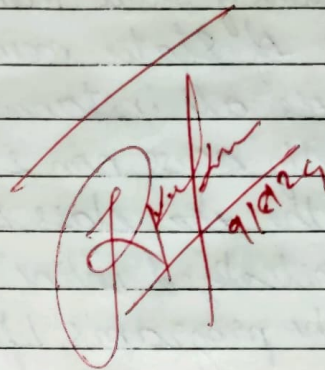
9) What are the advantages of using friend-function in C++?

→ The advantages of friend function are as follows:

- Access to private members: They can access private and protected members of class, allowing operations that require knowledge of the class.
- Enhanced encapsulation: Friend functions promote better encapsulation by controlling access to private data.



- Operator overloading: They facilitate operator overloading when the left operand isn't a class object.
- Inter-class relationships: They enable interactions between classes that need access to each other's private members.

 9/10/24

# OOC LABORATORY

## Lab Assignment : 01

Name : Ayush Kadali  
Roll No. : 54  
PRN : 1032232229  
Panel : B  
Batch : B2  
SY B.Tech CSE (AI & DS)

### Input Code:

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

class student
{
    public:
        string name;
        int rollNo,Class, division;
        string dob;
        int adharno;
        string bloodgroup;
        int contactno;
        string address;
        int telephoneno;
        static int count;

        static void total_students()
        {
            cout<<"Total student count "<<count << endl;
        }

        student()
        {
            name="Ayush";
            rollNo=54;
            Class=2;
            division=2;
            adharno=98467812;
```

```

        bloodgroup="B+";
        contactno=7588742;
        address="Pune";
        telephoneno=94266061;
        dob = "07.12.2005";
        count=count+1;
    }

    void putdata()
    {
        cout<<"Student Name is "<<name<<endl;
        cout<<"Student Roll No. is "<<rollno<<endl;
        cout<<"Student Class is "<<Class<<endl;
        cout<<"Student Division is "<<division<<endl;
        cout<<"Student Date of Birth is "<<dob<<endl;
        cout<<"Student Aadhar No. is "<<adharno<<endl;
        cout<<"Student Blood Group is "<<bloodgroup<<endl;
        cout<<"Student Contact No. is "<<contactno<<endl;
        cout<<"Student Address is "<<address<<endl;
        cout<<"Student Telephone No. is "<<telephoneno<<endl;
    }

};

int student::count;

int main()
{
    int i;
    student a, b;
    a.putdata();

    cout << "Putting 2 students (a & b)" << endl;
    student::total_students();
    return 0;
}

```



## OUTPUT

Student Name is Ayush

Student Roll No. is 54

Student Class is 2

Student Division is 2

Student Date of Birth is 07.12.2005

Student Aadhar No. is 98467812

Student Blood Group is B+

Student Contact No. is 7588742

Student Address is Pune

Student Telephone No. is 94266061

Putting 2 students (a & b)

Total student count 2