

Object Oriented Programming Lab

Assignment 4

Submitted by:

Navdeep Singh

19th August 2025

Roll No: 24124073

Group: 3

Branch: Information Technology

Year: 2nd Year

Practice Question to practice class and scope of static variable declared as member function or declared as globally !!

Code

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 class Student{
5     int id;
6     int marks;
7     static int count; // Static member inside class
8
9 public:
10    void display(){
11        cout <<"id "<< id << endl;
12        cout <<"marks : " << marks << endl;
13        count++;
14        cout << "Count: " << count << endl;
15    }
16
17    void setValues(int a , int m ){
18        id = a ;
19        marks = m ;
20    }
21 };
22
23 // Definition of static member
24 int Student::count = 0;
25
26 int main(){
27     Student s1;
28     s1.setValues(1 , 98) ;
29     s1.display();
30     s1.display();
31     s1.display();
32     return 0;
33 }
```

Sample Output

```
1 id 1
2 marks : 98
3 Count: 1
4 id 1
5 marks : 98
6 Count: 2
7 id 1
8 marks : 98
9 Count: 3
```

Q1. Write a C++ program to define a class named **BankAccount** that performs the following operations : i. Declare a static data member named **totalAccounts** to keep track of the total number of bank accounts created. ii. Declare a non-static data member named **accountNumber** to store the account number of each individual account. iii. Define a public member function named **setAccountNumber()** that: a. Accepts an account number as a parameter. b. Sets the **accountNumber** for the object. c. Increments the **totalAccounts** counter each time it is called. iv. Define a member function named **showTotalAccounts()** that: a. Displays the total number of accounts created by accessing the static variable.

Code

```
1 static int totalAccounts; // declaration of global static variable
2
3 class BankAccount{
4     int accountNo;
5
6     public :
7         void setAccountNumber(int no){
8             accountNo = no;
9             totalAccounts++;
10        }
11
12        void showTotalAccounts(){
13            cout<< "Total Account Number : " <<totalAccounts;
14        }
15 };
16
17
18 int main1(){
19     BankAccount b1;
20     b1.setAccountNumber(94);
21     b1.setAccountNumber(95);
22     b1.setAccountNumber(96);
23     b1.setAccountNumber(97);
24     b1.showTotalAccounts();
25 }
```

Sample Output

```
1 Total Account Number : 4
```

Q2. Write a C++ program that defines a class User to simulate user registration in a system, with the following requirements:

- Define a static data member nextID to keep track of the next available unique user ID (starting from 1000).
- Define a non-static data member userID to store the ID of each registered user.
- Create a member function registerUser() that:
 - Assigns the current nextID value to the userID of the object.
 - Increments nextID so the next user gets a new ID.
- Create a member function showUser() to display the userID of the object.
- Create another member function showNextID() that:
 - Displays the next user ID to be assigned (by accessing the static data member).

Code

```
1 class User{
2     public:
3         static int nextId; // static variable declared as data member
                           // but its value should be declared outside the function
4
5         int userID;
6         int temp = nextId;
7         User(){
8             userID = nextId;
9             nextId++;
10        }
11
12        void registerUser(){
13            userID = nextId;
14            nextId++;
15            cout<<"User Registered with ID : " <<userID <<endl;
16        }
17
18        void showUser(){
19            cout<<"User id of the current user : " <<userID <<endl;
20        }
21
22        void showNextId(){
23            cout<<"Next User id to be assigned to the user is : " <<
                           nextId <<endl;
24        }
25 };
26
27
28 int User :: nextId = 1000; // value initilzed of static data member
29
30 int main2(){
31     User s1;
32     s1.showUser();
33     s1.registerUser();
34     s1.registerUser();
35     s1.registerUser();
}
```

```
36     s1.showUser();  
37     s1.showNextId();  
38 }
```

Sample Output

```
1 User id of the current user : 1000  
2 User Registered with ID : 1001  
3 User Registered with ID : 1002  
4 User Registered with ID : 1003  
5 User id of the current user : 1003  
6 Next User id to be assigned to the user is : 1004
```