

# 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
```

## 1 Assignment 4 V1

**Q1. Write a C++ program to define a class named BankAccount that performs the following operations:**

- Declare a static data member named `totalAccounts` to keep track of the total number of bank accounts created.
- Declare a non-static data member named `accountNumber` to store the account number of each individual account.
- Define a public member function named `setAccountNumber()` that:
  - Accepts an account number as a parameter.
  - Sets the `accountNumber` for the object.
  - Increments the `totalAccounts` counter each time it is called.
- Define a member function named `showTotalAccounts()` that displays the total number of accounts created by accessing the static variable.

### Code

```
1 static int totalAccounts; // declartion 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
```

## 2 Assignment 4 V2

**Q3. Write a C++ program to define a class named `LibraryBook` to manage book registrations in a library. The class should perform the following:**

- Declare a static data member `totalBooks` to count how many books have been registered.
- Declare a non-static data member `bookID` to hold each book's unique ID.
- Create a member function `registerBook()` that:
  - Accepts a book ID as a parameter.
  - Assigns it to the current object's `bookID`.
  - Increments the static `totalBooks` counter.
- Define a member function `displayBook()` to display the book's ID.
- Define another non-static member function `displayTotalBooks()` that displays the current value of `totalBooks`.

In the `main()` function:

- Create at least four objects of the `LibraryBook` class.
- Register a book for each object using the `registerBook()` function, assigning a different book ID to each.
- Display the book ID of each object using the `displayBook()` function.
- Using any one of the objects, display the total number of books registered by calling the `displayTotalBooks()` function.

## Code

```
1 class LibraryBook {
2     static int totalBooks;    // static member
3     int bookID;              // non-static member
4
5     public:
6         // Register a book with given ID
7         void registerBook(int bookID) {
8             this->bookID = bookID;
9             totalBooks++;
10        }
11
12        // Display book ID
13        void displayBook() {
14            cout << "Book ID: " << bookID << endl;
15        }
16
17        // Display total number of books registered
18        void displayTotalBooks() {
19            cout << "Total Books: " << totalBooks << endl;
20        }
21 };
22
23 // Define and initialize static member
24 int LibraryBook::totalBooks = 0;
25
26 int main() {
27     // Create 4 objects
28     LibraryBook b1, b2, b3, b4;
29
30     // registering books
31     b1.registerBook(217);
32     b2.registerBook(318);
33     b3.registerBook(765);
34     b4.registerBook(654);
35
36     // Display each book ID
37     b1.displayBook();
38     b2.displayBook();
39     b3.displayBook();
40     b4.displayBook();
41
42     b1.displayTotalBooks();
43
44     return 0;
45 }
```

## Sample Output

```
1 Book ID: 217
2 Book ID: 318
3 Book ID: 765
4 Book ID: 654
5 Total Books: 4
```

#### Q4. Write a C++ program that defines a class `OnlineOrder` to manage customer orders with the following specifications:

- Define a static data member `orderCount` to track the number of orders.
- Define a non-static data member `orderNumber`.
- Create a non-static member function `placeOrder()` that:
  - Sets `orderNumber` to the current value of `orderCount` + 1.
  - Increments `orderCount`.
- Define another function `showOrder()` to display the `orderNumber`.
- Define a non-static member function `showOrderCount()` to display the current total order count using the static member.

In the `main()` function:

- Create at least four objects of the `OnlineOrder` class.
- Call the `placeOrder()` function for each object to simulate placing an order.
- Display the order details of each object using the `showOrder()` function.
- Using any one of the objects, display the total number of orders placed by calling the `showOrderCount()` function.

#### Code

```
1 class OnlineOrder {
2     static int orderCount;    // Static member to track total orders
3     int orderNumber;         // Each object's order number
4
5 public:
6     // Place an order
7     void placeOrder() {
8         orderNumber = orderCount + 1;    // Set order number
9         orderCount++;                    // Increment total orders
10    }
11
12    // Show order number
13    void showOrder() {
14        cout << "Order Number: " << orderNumber << endl;
15    }
16
17    // Show total order count
18    void showOrderCount() {
19        cout << "Total Orders: " << orderCount << endl;
20    }
21 };
22
23 // Initialize static member
24 int OnlineOrder::orderCount = 0;
25
26 int main() {
27     // Create 4 objects
```

```
28     OnlineOrder o1, o2, o3, o4;
29
30     // Place orders
31     o1.placeOrder();
32     o2.placeOrder();
33     o3.placeOrder();
34     o4.placeOrder();
35
36     // Show order details
37     o1.showOrder();
38     o2.showOrder();
39     o3.showOrder();
40     o4.showOrder();
41
42     // Show total orders using one object
43     o1.showOrderCount();
44
45     return 0;
46 }
```

## Sample Output

```
1 Order Number: 1
2 Order Number: 2
3 Order Number: 3
4 Order Number: 4
5 Total Orders: 4
```

## 3 Assignment 4 V3

**Q5. Write a C++ program to define a class named Student that assigns roll numbers to students automatically:**

- Declare a static data member named `nextRoll` initialized to 1.
- Declare a non-static data member named `rollNo`.
- Define a public member function named `assignRoll()` that:
  - Assigns `nextRoll` to `rollNo`.
  - Increments `nextRoll`.
- Define a member function named `showRoll()` that displays the roll number of the student.
- Define a member function named `showNextRoll()` that displays the next roll number to be assigned by accessing the static variable.
- In the `main()` function:
  - Create at least five **Student** objects.
  - Call the `assignRoll()` function for each object to assign a roll number.
  - Display each student's roll number using the `showRoll()` function.



- Using any one object, display the next roll number using the `showNextRoll()` function.

## Code

```
1 class Student {
2 private:
3     static int nextRoll; // static data member
4     int rollNo;          // non-static data member
5
6 public:
7     void assignRoll() {
8         rollNo = nextRoll;
9         nextRoll++;
10    }
11
12    void showRoll() {
13        cout << "Student Roll Number: " << rollNo << endl;
14    }
15
16    void showNextRoll() {
17        cout << "Next Roll Number will be: " << nextRoll << endl;
18    }
19 };
20
21 // initializing static member
22 int Student::nextRoll = 1;
23
24 int main() {
25     // creating objects
26     Student s1, s2, s3, s4, s5;
27
28     // assigning roll numbers
29     s1.assignRoll();
30     s2.assignRoll();
31     s3.assignRoll();
32     s4.assignRoll();
33     s5.assignRoll();
34
35     // displaying roll numbers
36     s1.showRoll();
37     s2.showRoll();
38     s3.showRoll();
39     s4.showRoll();
40     s5.showRoll();
41
42     // showing next roll number
43     s3.showNextRoll();
44
45     return 0;
46 }
```

## Sample Output

```
1 Student Roll Number: 1
2 Student Roll Number: 2
```

```
3 Student Roll Number: 3
4 Student Roll Number: 4
5 Student Roll Number: 5
6 Next Roll Number will be: 6
```

### Q6. Write a C++ program to define a class named **Product** to simulate product registration in an inventory system:

- Declare a static data member named `nextProductID` starting from 500.
- Declare a non-static data member named `productID` to store the unique ID of each product.
- Define a member function named `registerProduct()` that:
  - Assigns `nextProductID` to the object's `productID`.
  - Increments `nextProductID` for the next product registration.
- Define a member function named `showProductID()` that displays the product's assigned ID.
- Define a member function named `showUpcomingID()` that displays the next product ID to be assigned by accessing the static variable.
- In the `main()` function:
  - Create at least four **Product** objects.
  - Call the `registerProduct()` function for each object to assign an ID.
  - Display each product's ID using the `showProductID()` function.
  - Display the upcoming product ID using the `showUpcomingID()` function.

### Code

```
1 class Product {
2 private:
3     static int nextProductID; // static data member
4     int productID;           // non-static data member
5
6 public:
7     void registerProduct() {
8         productID = nextProductID;
9         nextProductID++;
10    }
11
12    void showProductID() {
13        cout << "Product ID: " << productID << endl;
14    }
15
16    void showUpcomingID() {
17        cout << "Next Product ID will be: " << nextProductID << endl;
18    }
19 };
20
```

```
21 // initializing static member
22 int Product::nextProductID = 500;
23
24 int main() {
25     // creating objects
26     Product p1, p2, p3, p4;
27
28     // registering products
29     p1.registerProduct();
30     p2.registerProduct();
31     p3.registerProduct();
32     p4.registerProduct();
33
34     // displaying product IDs
35     p1.showProductID();
36     p2.showProductID();
37     p3.showProductID();
38     p4.showProductID();
39
40     // showing upcoming ID
41     p2.showUpcomingID();
42
43     return 0;
44 }
```

## Sample Output

```
1 Product ID: 500
2 Product ID: 501
3 Product ID: 502
4 Product ID: 503
5 Next Product ID will be: 504
```

\*\*\*\*\* END OF ASSIGNMENT \*\*\*\*\*