CodeWithHarry

Static Data Members & Methods in C++ OOPS | C++ Tutorials for Beginners #24

▶

Overview   Q&A   Downloads   Announcements

Static Data Members & Methods in C++ OOPS | C++ Tutorials for Beginners #24

In this tutorial, we will discuss static data members and methods in C++

Static Data Members in C++

When a static data member is created, there is only a single copy of the data member which is shared between all the objects of the class. As we have discussed in our previous lecture that if the data members are not static then every object has an individual copy of the data member and it is not shared.

Static Methods in C++

When a static method is created, they become independent of any object and class. Static methods can only access static data members and static methods. Static methods can only be accessed using the scope

resolution operator. An example program is shown below to demonstrate static data members and static methods in C++.

```cpp
class Employee
{
    int id;
    static int count;

public:
    void setData(void)
    {
        cout << "Enter the id" << endl;
        cin >> id;
        count++;
    }
    void getData(void)
    {
        cout << "The id of this employee is " << id << " and this is employee number " << co
    }

    static void getCount(void){
        // cout<<id; // throws an error
        cout<<"The value of count is "<<count<<endl;
    }
};
```

Code Snippet 1: Employee Class

As shown in Code Snippet 1, we created an employee class that has integer "id" variable and "count" static integer variable as private class members; and "setData" void function, "getData" void function, and "getCount" static void function as public class members. These functions are explained below.

We have defined a "setData" function. This function will take input for "id" from the user at runtime and increment in the count. The value of the counter will be incremented by one every time this function will run.

We have defined a "getData" function. This function will print the values of the variables "id" and "count".

We have defined a static "getCount" function. This function will print the value of the variable count". The main thing to note here is that "getCount" function is static, so if we try to access any data members or member functions which are not static the compiler will throw an error.

```cpp
// Count is the static data member of class Employee
int Employee::count; // Default value is 0


int main()
{
    Employee harry, rohan, lovish;
    // harry.id = 1;
    // harry.count=1; // cannot do this as id and count are private

    harry.setData();
    harry.getData();
    Employee::getCount();

    rohan.setData();
    rohan.getData();
```

```cpp
    Employee::getCount();

    lovish.setData();
    lovish.getData();
    Employee::getCount();

    return 0;
}
```
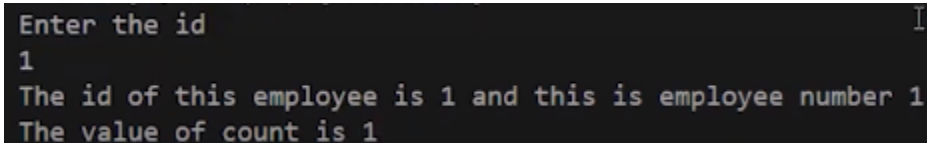
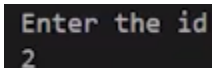*Code Snippet 2: main Program*

As shown in Code Snippet 2:

- The count variable is declared whose default value is "0".
- Then we created objects "harry", "rohan", and "lovish" of the employee data type
- The functions "setData", "getData" are called by the object "harry", the function "getCount" is called by using class name and scope resolution operator because it is a static method.
- The functions "setData", "getData" are called by the object "rohan", the function "getCount" is called by using class name and scope resolution operator because it is a static method.
- The functions "setData", "getData" are called by the object "lovish", the function "getCount" is called by using class name and scope resolution operator because it is a static method.

The output of the following program is shown in figures 1 and 2.

```
Enter the id                                                    I
1
The id of this employee is 1 and this is employee number 1
The value of count is 1
```

Figure 1: Program Output 1

```
Enter the id
2
```

```
The id of this employee is 2 and this is employee number 2
The value of count is 2                                    I
Enter the id
3
The id of this employee is 3 and this is employee number 3
The value of count is 3
```

Figure 2: Program Output 2

As shown in figures 1 and 2, for the "harry" object we entered the ID "1"; for the "rohan" object we entered the ID "2"; and for the "lovish" object we entered the ID "3". The Output of the program has displayed the ID and the count of each employee.

Code as described/written in the video

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

class Employee
{
    int id;
    static int count;

public:
    void setData(void)
    {
        cout << "Enter the id" << endl;
        cin >> id;
        count++;
    }
    void getData(void)
```

```cpp
    {
        cout << "The id of this employee is " << id << " and this is employee number " << co
    }

    static void getCount(void){
        // cout<<id; // throws an error
        cout<<"The value of count is "<<count<<endl;
    }
};

// Count is the static data member of class Employee
int Employee::count; // Default value is 0

int main()
{
    Employee harry, rohan, lovish;
    // harry.id = 1;
    // harry.count=1; // cannot do this as id and count are private

    harry.setData();
    harry.getData();
    Employee::getCount();

    rohan.setData();
    rohan.getData();
    Employee::getCount();
```

```cpp
    lovish.setData();
    lovish.getData();
    Employee::getCount();

    return 0;
}
```

Previous

Next