**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++.

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 " << count << endl;

}

static void getCount(void){

// cout<<id; // throws an error

cout<<"The value of count is "<<count<<endl;

}

};

***Code Snippet 1: Employee Class***

// 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();

lovish.setData();

lovish.getData();

Employee::getCount();

return 0;

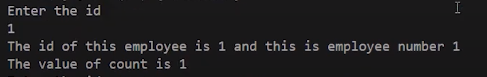
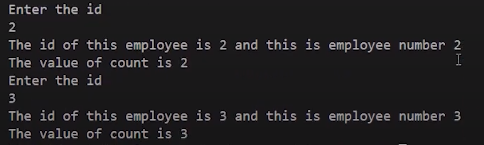
}

***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.

******