 **Static data members**

Static data members are class members that are declared using the static keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only one copy of the static member. A static member is shared by all objects of the class.

Declaration

static data\_type member\_name;

Defining the static data member:It should be defined outside of the class

following this syntax:

data\_type class\_name :: member\_name =value;

Static data members are used to store value common to the whole class. The static data member differs from an ordinary data member in the following ways :

(i) Only a single copy of the static data member is used by all the objects.

(ii) It can be used within the class but its lifetime is the whole program.

For making a data member static, we require :

(a) Declare it within the class.

(b) Define it outside the class.

 suppose in a function there are 2 variables, one is a normal variable and the other one is a static variable. The normal variable is created when the function is called and its scope is limited. While the static variable is created once and destroyed at the end of the program. These variables have a lifetime throughout the program.

#include <iostream>

 using namespace std;

void Test(){

 static int x = 1;

    x = ++x;

    int y = 1;

   y = ++y;

   cout<<"x = "<<x<<"n";

   cout<<"y = "<<y<<"n";

}

int main()

{

   Test();

   Test();

    return 0;

}

Output:

**Static member functions:**

A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::. A static member function can only access static data member, other static member functions and any other functions from outside the class.

We can access a static member function with class name, by using following syntax:

class\_name:: function\_name(parameter);

·        A static member function can access only the static members of a class.

·        In C++, a static member function fifers from the other member functions in the following ways:

   (i) Only static members (functions or variables) of the same class can be accessed by a static member function.

  (ii) It is called by using the **name of the class rather than an object**

For example:

        student::showcount();

**Here is the example static data member is accessing through the static member function:**

#include <iostream>

using namespace std;

class Demo

{

          private:

                      static int X;

          public:

                      static void fun()

                      {

                                  cout <<"Value of X: " << X << endl;

                      }

};

//defining

int Demo :: X =10;

int main()

{

          Demo X;

          X.fun();

          return 0;

}

**Here is the example to show the difference between static data members and non static member**

class base

{

public:

int x;

static int y;

void printx()

{

cout<<x<<endl;

cout<<y<<endl;

static void printy()

{

cout<<y<<endl;

}

};

int base::y;

int main()

{

base b1;

b1.x=10;

base::y=30;

base b2;

b2.x=20

base::y=40;

b1.printx();

base::printy();

b2.printy();

base::printx();

return 0;

}

Just like static member variables we have static member functions that are used for a specific purpose. To create a static member function we need to use the static keyword while declaring the function. Since static member variables are class properties and not object properties, to access them we need to use the class name instead of the object name.

**Properties of static member functions:**

A static function can only access other static variables or functions present in the same class

Static member functions are called using the class name. Syntax- class\_name::function\_name( )

Let’s consider a classic example to understand the concept of static member functions in detail. In this example, we will understand all the concepts related to static member functions.

#include <iostream>

using namespace std;

class Example{

   static int Number;

   int n;

   public:

   void set\_n(){

       n = ++Number;

   }

   void show\_n(){

       cout<<"value of n = "<<n<<endl;

   }

   static void show\_Number(){

  cout<<"value of Number = "<<Number<<endl;

   }

};

int Example:: Number;

int main()

{

Example example1, example2;

 example1.set\_n();

example2.set\_n();

 example1.show\_n();

 example2.show\_n();

   Example::show\_Number();

   return 0;

}

Any changes in the static data member through one member function will reflect in all other object’s member functions.

**Summary**

* In this lecture we have discussed about Static data members.
* We have discussed static member functions.

**Frequently Asked Questions**

Some of the most frequently asked questions are:

**Q1 What are Static data members**?

Static data members are class members that are declared using the static keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only one copy of the static member. A static member is shared by all objects of the class.

**Q2 What are static member functions?**

A static member function can be called even if no objects of the class exist and the static functions are accessed using only the class name and the scope resolution operator ::. A static member function can only access static data member, other static member functions and any other functions from outside the class.

**Q3 How static data members are different from non static data members?**

In case of static data members only one copy of the data member is created for the entire class and is shared by all the objects.However non static data members belong to the objects .and there will be different memories allocated.