



Static Keyword in C++



Prerequisite: [Static variables in C](#)

The static keyword has different meanings when used with different types. We can use static keywords with:

1. **Static Variables:** Variables in a function, Variables in a class
2. **Static Members of Class:** Class objects and Functions in a class Let us now look at each one of these uses of static in detail.

Static Variables

Static variables in a Function: When a variable is declared as static, space for it gets allocated for the lifetime of the program. Even if the function is called multiple times, space for the static variable is allocated only once and the value of the variable in the previous call gets carried through the next function call. This is useful for implementing [coroutines in C/C++](#) or any other application where the previous state of function needs to be stored.

C++



C++ Data Types C++ Input/Output C++ Arrays C++ Pointers C++ OOPs C++ STL C++ Interview Questions

```
2 // the use of static Static
3 // variables in a Function
4 #include <iostream>
5 #include <string>
6 using namespace std;
7
8 void demo()
9 {
10     // static variable
11     int count = 0;
```

```
12     cout << count << " ";
13
14     // value is updated and
15     // will be carried to next
16     // function calls
17     count++;
18 }
19
20 int main()
21 {
22     for (int i = 0; i < 5; i++)
23         demo();
24     return 0;
25 }
26
```

Output

```
0 1 2 3 4
```

You can see in the above program that the variable count is declared static. So, its value is carried through the function calls. The variable count is not getting initialized every time the function is called. As a side note, [Java doesn't allow static local variables in functions](#).



Static variables in a class: As the variables declared as static are initialized only once as they are allocated space in separate static storage so, the static variables in a class are shared by the objects. There can not be multiple copies of the same static variables for different objects. Also because of this reason static variables can not be initialized using constructors.

C++

```
// C++ program to demonstrate static
// variables inside a class

#include <iostream>
using namespace std;

class GfG {
public:
    static int i;

    GfG(){
        // Do nothing
    };
};

int main()
{
    GfG obj1;
    GfG obj2;
    obj1.i = 2;
    obj2.i = 3;

    // prints value of i
```

```
    cout << obj1.i << " " << obj2.i;  
}
```

Output

```
undefined reference to `GfG::i'  
collect2: error: ld returned 1 exit status
```

You can see in the above program that we have tried to create multiple copies of the static variable `i` for multiple objects. But this didn't happen. So, a static variable inside a class should be initialized explicitly by the user using the class name and scope resolution operator outside the class as shown below:

C++

```
// C++ program to demonstrate static  
// variables inside a class  
  
#include <iostream>  
using namespace std;  
  
class GfG {  
public:  
    static int i;  
  
    GfG(){  
        // Do nothing  
    };  
};  
  
int GfG::i = 1;  
  
int main()  
{  
    GfG obj;  
    // prints value of i  
    cout << obj.i;  
}
```

Output

1

Static Members of Class

Class objects as static: Just like variables, objects also when declared as static have a scope till the lifetime of the program. Consider the below program where the object is non-static.

C++

```
// CPP program to illustrate
// when not using static keyword
#include <iostream>
using namespace std;

class GfG {
    int i;

public:
    GfG()
    {
        i = 0;
        cout << "Inside Constructor\n";
    }
    ~GfG() { cout << "Inside Destructor\n"; }
};

int main()
{
    int x = 0;
    if (x == 0) {
        GfG obj;
    }
    cout << "End of main\n";
}
```

Output

Inside Constructor

Inside Destructor

End of main

In the above program, the object is declared inside the if block as non-static. So, the scope of a variable is inside the if block only. So when the object is created the constructor is invoked and soon as the control of if block gets over the destructor is invoked as the scope of the object is inside the if block only where it is declared. Let us now see the change in output if we declare the object as static.

C++

```
// CPP program to illustrate
// class objects as static
#include <iostream>
using namespace std;

class GfG {
    int i = 0;

public:
    GfG()
    {
        i = 0;
        cout << "Inside Constructor\n";
    }

    ~GfG() { cout << "Inside Destructor\n"; }
};

int main()
{
    int x = 0;
    if (x == 0) {
        static GfG obj;
    }
    cout << "End of main\n";
}
```

Output

```
Inside Constructor
End of main
Inside Destructor
```

You can clearly see the change in output. Now the destructor is invoked after the end of the main. This happened because the scope of static objects is throughout the lifetime of the program.

Static functions in a class: Just like the static data members or static variables inside the class, static member functions also do not depend on the object of the class. We are allowed to invoke a static member function using the object and the '.' operator but it is recommended to invoke the static members using the class name and the scope resolution operator. **Static member functions are allowed to access only the static data members or other static member functions**, they can not access the non-static data members or member functions of the class.

C++

```
// C++ program to demonstrate static
// member function in a class
#include <iostream>
using namespace std;

class GfG {
public:
    // static member function
    static void printMsg() { cout << "Welcome to GfG!"; }
};

// main function
int main()
{
    // invoking a static member function
    GfG::printMsg();
}
```

Output

```
Welcome to GfG!
```

Related Articles:

- [Quiz on Static Keyword](#)
- [Static data members in C++](#)
- [When are static objects destroyed?](#)
- [Interesting facts about static member functions](#)
- [Can static functions be virtual?](#)
- [Comparison of static keyword in C++ and Java](#)
- [Static functions in C](#)

Your options after **clearing GATE Examinations** can be:

1. Go for higher studies in prestigious IITs
2. Apply for job roles in PSUs or MNCs
3. Specialize further in AI or Cybersecurity
4. Take up teaching roles
5. And much more!

Sounds like something you wish to learn more about? [Register for Free GATE Counselling session](#) with our experts and they shall guide you in the right direction.

Also get assured GfG T-Shirts if you attend the counselling. [Register Now.](#)

NOTE: This service is exclusively for the students in Delhi/NCR Region as the classroom program for GATE is in our Noida Center only. We are soon coming up with more all across India!

Last Updated : 16 Nov, 2023

 283 

[< Previous](#)[Next >](#)[Storage Classes in C++ with Examples](#)[C++ Data Types](#)[Share your thoughts in the comments](#)[Add Your Comment](#)

Similar Reads

[Comparison of static keyword in C++ and Java](#)[C++ Program to Show Use of This Keyword in Class](#)[How to fix auto keyword error in Dev-C++](#)[delete keyword in C++](#)[Const keyword in C++](#)[Using Keyword in C++ STL](#)[Use of explicit keyword in C++](#)[Function overloading and const keyword](#)[C++ mutable keyword](#)[How to Use Const Keyword in C++?](#)**GeeksforGeeks****Article Tags :** [C++](#)**Practice Tags :** [CPP](#)**Additional Information**



Sanchhaya Education Private Limited



A-143, 9th Floor, Sovereign Corporate
Tower, Sector-136, Noida, Uttar Pradesh -
201305



Company

[About Us](#)[Legal](#)[Careers](#)[In Media](#)[Contact Us](#)[Advertise with us](#)[GFG Corporate Solution](#)[Placement Training Program](#)

Explore

[Hack-A-Thons](#)[GfG Weekly Contest](#)[DSA in JAVA/C++](#)[Master System Design](#)[Master CP](#)[GeeksforGeeks Videos](#)[Geeks Community](#)

Languages

[Python](#)[Java](#)[C++](#)[PHP](#)[GoLang](#)[SQL](#)

DSA

[Data Structures](#)[Algorithms](#)[DSA for Beginners](#)[Basic DSA Problems](#)[DSA Roadmap](#)[Top 100 DSA Interview Problems](#)

[R Language](#)[Android Tutorial](#)[Tutorials Archive](#)

Data Science & ML

[Data Science With Python](#)[Data Science For Beginner](#)[Machine Learning Tutorial](#)[ML Maths](#)[Data Visualisation Tutorial](#)[Pandas Tutorial](#)[NumPy Tutorial](#)[NLP Tutorial](#)[Deep Learning Tutorial](#)

Python

[Python Programming Examples](#)[Python Projects](#)[Python Tkinter](#)[Web Scraping](#)[OpenCV Python Tutorial](#)[Python Interview Question](#)

DevOps

[Git](#)[AWS](#)[Docker](#)[Kubernetes](#)[Azure](#)[GCP](#)[DevOps Roadmap](#)

System Design

[High Level Design](#)[DSA Roadmap by Sandeep Jain](#)[All Cheat Sheets](#)

HTML & CSS

[HTML](#)[CSS](#)[Web Templates](#)[CSS Frameworks](#)[Bootstrap](#)[Tailwind CSS](#)[SASS](#)[LESS](#)[Web Design](#)[Django Tutorial](#)

Computer Science

[Operating Systems](#)[Computer Network](#)[Database Management System](#)[Software Engineering](#)[Digital Logic Design](#)[Engineering Maths](#)

Competitive Programming

[Top DS or Algo for CP](#)[Top 50 Tree](#)[Top 50 Graph](#)[Top 50 Array](#)[Top 50 String](#)[Top 50 DP](#)[Top 15 Websites for CP](#)

JavaScript

[JavaScript Examples](#)

[Low Level Design](#)[UML Diagrams](#)[Interview Guide](#)[Design Patterns](#)[OOAD](#)[System Design Bootcamp](#)[Interview Questions](#)[TypeScript](#)[ReactJS](#)[NextJS](#)[AngularJS](#)[NodeJS](#)[Lodash](#)[Web Browser](#)

Preparation Corner

[Company-Wise Recruitment Process](#)[Resume Templates](#)[Aptitude Preparation](#)[Puzzles](#)[Company-Wise Preparation](#)

School Subjects

[Mathematics](#)[Physics](#)[Chemistry](#)[Biology](#)[Social Science](#)[English Grammar](#)[World GK](#)

Management & Finance

[Management](#)[HR Management](#)[Finance](#)[Income Tax](#)[Organisational Behaviour](#)[Marketing](#)

Free Online Tools

[Typing Test](#)[Image Editor](#)[Code Formatters](#)[Code Converters](#)[Currency Converter](#)[Random Number Generator](#)[Random Password Generator](#)

More Tutorials

[Software Development](#)[Software Testing](#)[Product Management](#)[SAP](#)[SEO - Search Engine Optimization](#)[Linux](#)[Excel](#)

GeeksforGeeks Videos

[DSA](#)[Python](#)[Java](#)[C++](#)[Data Science](#)[CS Subjects](#)

