

[Courses](#)[Login](#)[Suggest an Article](#)

C# | Inheritance

Inheritance is an important pillar of OOP(Object Oriented Programming). It is the mechanism in C# by which one class is allowed to inherit the features(fields and methods) of another class.

Important terminology:

- **Super Class:** The class whose features are inherited is known as super class(or a base class or a parent class).
- **Sub Class:** The class that inherits the other class is known as subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability:** Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

How to use inheritance

The symbol used for inheritance is :.

Syntax:

```
class derived-class : base-class
{
    // methods and fields
    .
```



```
}
```

Example: In below example of inheritance, class GFG is a base class, class GeeksforGeeks is a derived class which extends GFG class and class Sudo is a driver class to run program.

```
// C# program to illustrate the
// concept of inheritance
using System;
namespace ConsoleApplication1 {

// Base class
class GFG {

// data members
public string name;
public string subject;

// public method of base class
public void readers(string name, string subject)
{
    this.name = name;
    this.subject = subject;
    Console.WriteLine("Myself: " + name);
    Console.WriteLine("My Favorite Subject is: " + subject);
}

}

// inheriting the GFG class using :
class GeeksforGeeks : GFG {

// constructor of derived class
public GeeksforGeeks()
{
    Console.WriteLine("GeeksforGeeks");
}

}
```



```
// Driver class
class Sudo {

    // Main Method
    static void Main(string[] args)
    {

        // creating object of derived class
        GeeksforGeeks g = new GeeksforGeeks();

        // calling the method of base class
        // using the derived class object
        g.readers("Kirti", "C#");
    }
}
```

Output:

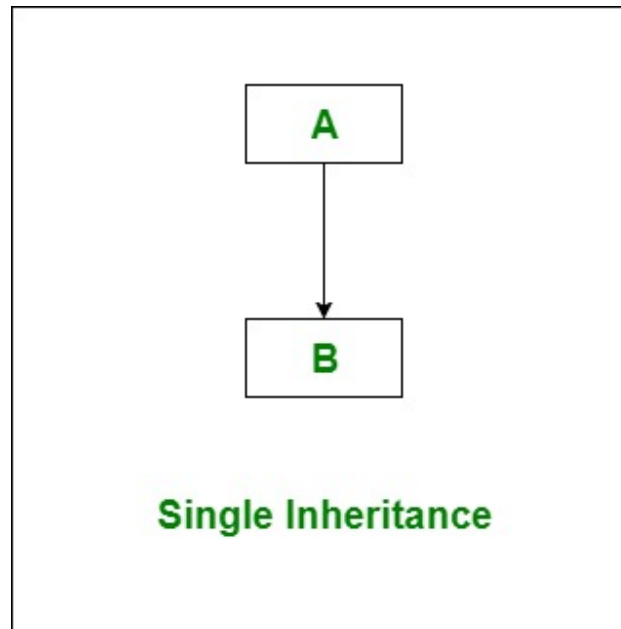
```
GeeksforGeeks
Myself: Kirti
My Favorite Subject is: C#
```

Types of Inheritance in C#

Below are the different types of inheritance which is supported by C# in different combinations.

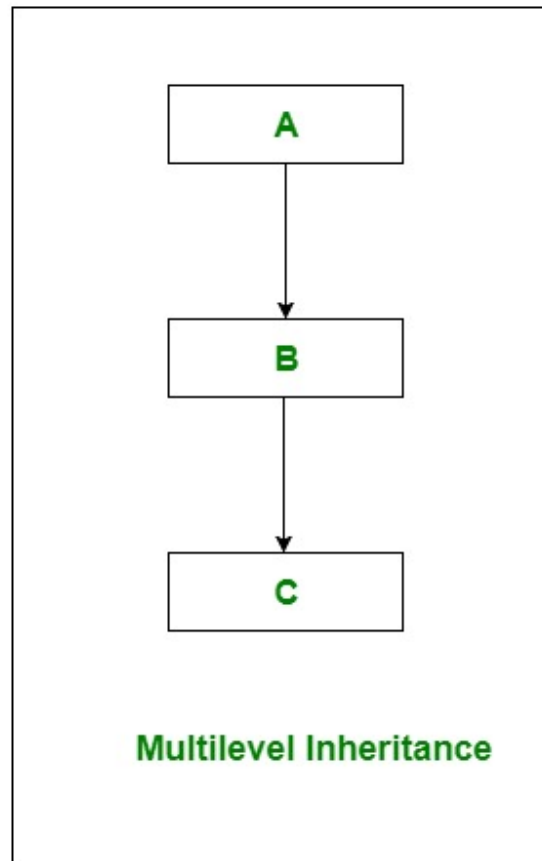
1. **Single Inheritance:** In single inheritance, subclasses inherit the features of one superclass. In image below, the class A serves as a base class for the derived class B.





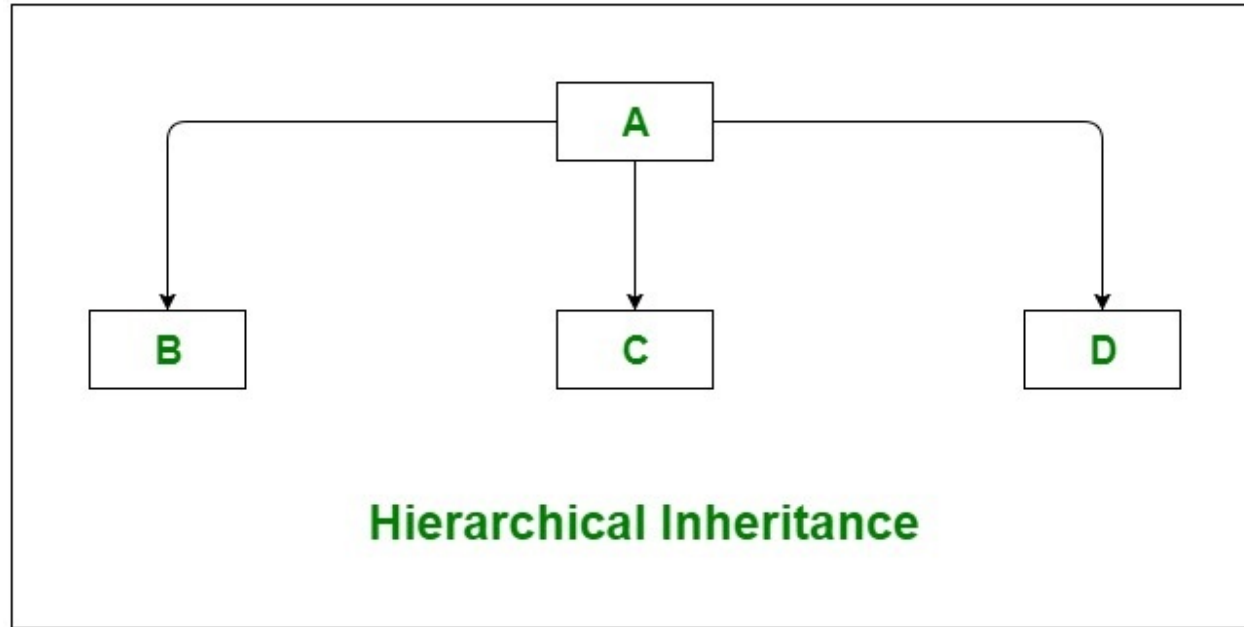
2. **Multilevel Inheritance:** In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class. In below image, class A serves as a base class for the derived class B, which in turn serves as a base class for the derived class C.



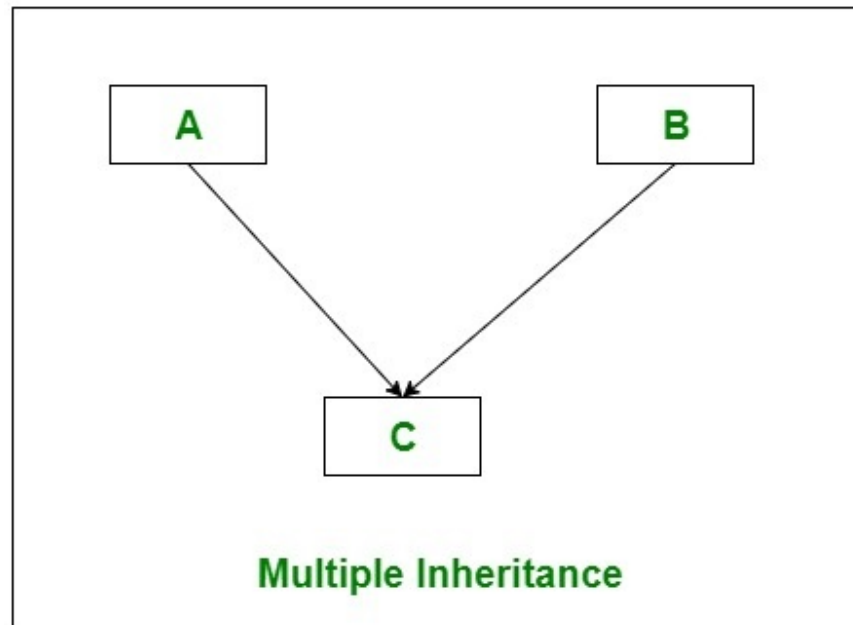


3. **Hierarchical Inheritance:** In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one subclass. In below image, class A serves as a base class for the derived class B, C, and D.

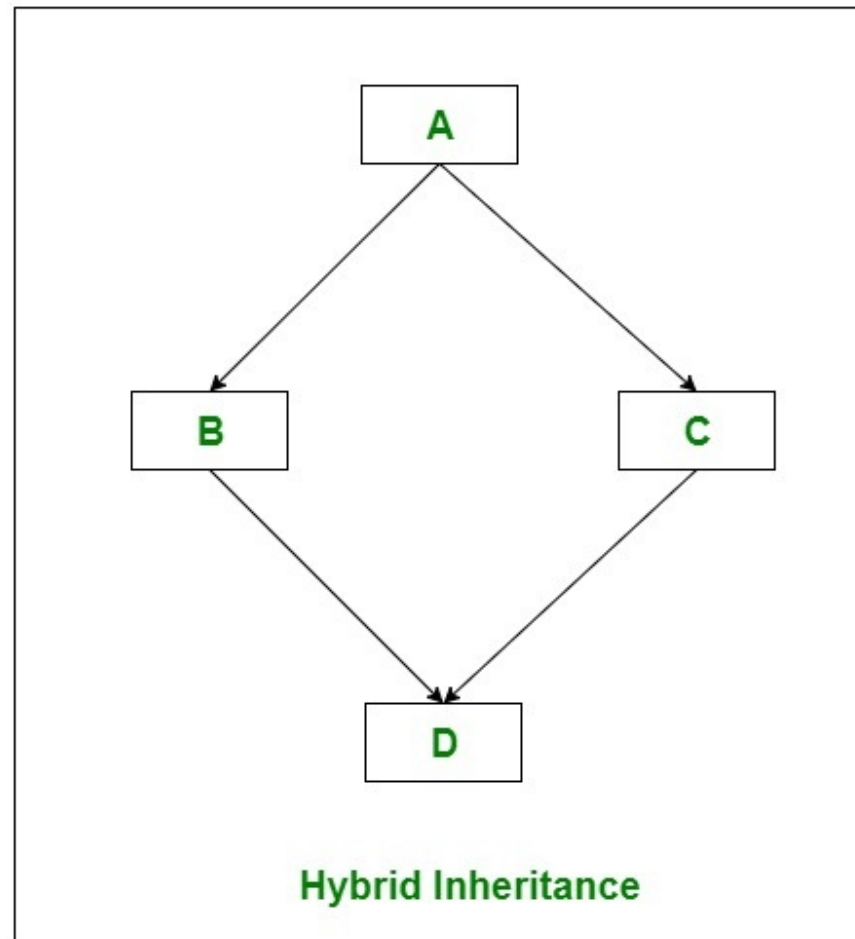




4. **Multiple Inheritance(Through Interfaces):** In Multiple inheritance, one class can have more than one superclass and inherit features from all parent classes. Please note that **C# does not support multiple inheritance** with classes. In C#, we can achieve multiple inheritance only through Interfaces. In the image below, Class C is derived from interface A and B.



5. **Hybrid Inheritance(Through Interfaces):** It is a mix of two or more of the above types of inheritance. Since C# doesn't support multiple inheritance with classes, the hybrid inheritance is also not possible with classes. In C#, we can achieve hybrid inheritance only through Interfaces.



Important facts about inheritance in C#

- **Default Superclass:** Except Object class, which has no superclass, every class has one and only one direct superclass(single inheritance). In the absence of any other explicit superclass, every class is implicitly a subclass of Object class.
- **Superclass can only be one:** A superclass can have any number of subclasses. But a subclass can have only **one** superclass. This is because C# does not support multiple inheritance with classes. Although with interfaces, multiple inheritance is

supported by C#.

- **Inheriting Constructors:** A subclass inherits all the members (fields, methods) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.
- **Private member inheritance:** A subclass does not inherit the private members of its parent class. However, if the superclass has properties(get and set methods) for accessing its private fields, then a subclass can inherit.

Recommended Posts:

[C# | Inheritance in Constructors](#)

[C# | Inheritance in interfaces](#)

[C# | Multilevel Inheritance](#)

[C# | Multiple inheritance using interfaces](#)

[MathF.Atan2\(\) Method in C# with Examples](#)

[MathF.Cos\(\) Method in C# with Examples](#)

[MathF.Asin\(\) Method in C# with Examples](#)

[MathF.Ceiling\(\) Method in C# with Examples](#)

[How to Extract filename from a given path in C#](#)

[MathF.Tanh\(\) Method in C# with Examples](#)

[MathF.Acos\(\) Method in C# with Examples](#)

[MathF.Cosh\(\) Method in C# with Examples](#)

[MathF.Sinh\(\) Method in C# with Examples](#)

[MathF.Sqrt\(\) Method in C# with Examples](#)



Kirti Mangal

Check out this Author's [contributed articles](#).



If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.



Article Tags : [C#](#) [CSharp-Inheritance](#) [CSharp-OOP](#)



2

☐ To-do ☐ Done

2

Based on 1 vote(s)

[Feedback/ Suggest Improvement](#)[Add Notes](#)[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)[Share this post!](#)

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved

