Difference between Abstract Class and Interface in C#

An [abstract class](https://www.geeksforgeeks.org/c-abstract-classes/) is a way to achieve the abstraction in C#. An Abstract class is never intended to be instantiated directly. This class must contain at least one abstract method, which is marked by the keyword or modifier *abstract*in the class definition. The Abstract classes are typically used to define a base class in the class hierarchy.

**Example:**

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| // C# program to illustrate the  // concept of abstract class  using System;    // abstract class 'G'  public abstract class G {        // abstract method 'gfg1()'      public abstract void gfg1();  }    // class 'G' inherit  // in child class 'G1'  public class G1 : G {        // abstract method 'gfg1()'      // declare here with      // 'override' keyword      public override void gfg1()      {          Console.WriteLine("Class name is G1");      }  }    // class 'G' inherit in  // another child class 'G2'  public class G2 : G {        // same as the previous class      public override void gfg1()      {          Console.WriteLine("Class name is  G2");      }  }    // Driver Class  public class main\_method {        // Main Method      public static void Main()      {            // 'obj' is object of class          // 'G' class '          // G' cannot          // be instantiate          G obj;            // instantiate class 'G1'          obj = new G1();            // call 'gfg1()' of class 'G1'          obj.gfg1();            // instantiate class 'G2'          obj = new G2();            // call 'gfg1()' of class 'G2'          obj.gfg1();      }  } |

**Output :**

Class name is G1

Class name is G2

Like a class, [Interface](https://www.geeksforgeeks.org/c-interface/)can have methods, properties, events, and indexers as its members. But interfaces will contain only the declaration of the members. The implementation of interface’s members will be given by the class who implements the interface implicitly or explicitly.

**Example:**

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| // C# program to illustrate the  // concept of interface  using System;    // A simple interface  interface interface1 {        // method having only declaration      // not definition      void show();  }    // A class that implements the interface.  class MyClass : interface1 {        // providing the body part of function      public void show()      {          Console.WriteLine("Welcome to GeeksforGeeks!!!");      }        // Main Method      public static void Main(String[] args)      {            // Creating object          MyClass obj1 = new MyClass();            // calling method          obj1.show();      }  } |

**Output:**

Welcome to GeeksforGeeks!!!

**Difference between Abstract Class and Interface**

|  |  |
| --- | --- |
| **ABSTRACT CLASS** | **INTERFACE** |
| It contains both declaration and definition part. | It contains only a declaration part. |
| Multiple inheritance is not achieved by abstract class. | Multiple inheritance is achieved by interface. |
| It contain [constructor](https://www.geeksforgeeks.org/c-sharp-constructors/). | It does not contain [constructor](https://www.geeksforgeeks.org/c-sharp-constructors/). |
| It can contain static members. | It does not contain static members. |
| It can contain different types of access modifiers like public, private, protected etc. | It only contains public access modifier because everything in the interface is public. |
| The performance of an abstract class is fast. | The performance of interface is slow because it requires time to search actual method in the corresponding class. |
| It is used to implement the core identity of class. | It is used to implement peripheral abilities of class. |
| A class can only use one abstract class. | A class can use multiple interface. |
| If many implementations are of the same kind and use common behavior, then it is superior to use abstract class. | If many implementations only share methods, then it is superior to use Interface. |
| Abstract class can contain methods, fields, constants, etc. | Interface can only contain methods . |
| It can be fully, partially or not implemented. | It should be fully implemented. |