

C# | Stack Class

 [geeksforgeeks.org/c-sharp-stack-class](https://www.geeksforgeeks.org/c-sharp-stack-class)

November 6,
2018

Stack represents a **last-in, first out** collection of object. It is used when you need a last-in, first-out access to items. When you add an item in the list, it is called **pushing** the item and when you remove it, it is called **popping** the item. This class comes under **System.Collections** namespace.

Characteristics of Stack Class:

- The capacity of a Stack is the number of elements the Stack can hold. As elements are added to a Stack, the capacity is automatically increased as required through reallocation.
- If Count is less than the capacity of the stack, Push is an $O(1)$ operation. If the capacity needs to be increased to accommodate the new element, Push becomes an $O(n)$ operation, where n is Count. Pop is an $O(1)$ operation.
- Stack accepts null as a valid value and allows duplicate elements.

Constructors

Constructor	Description
<u>Stack()</u>	Initializes a new instance of the Stack class that is empty and has the default initial capacity.
<u>Stack(ICollection)</u>	Initializes a new instance of the Stack class that contains elements copied from the specified collection and has the same initial capacity as the number of elements copied.
<u>Stack(Int32)</u>	Initializes a new instance of the Stack class that is empty and has the specified initial capacity or the default initial capacity, whichever is greater.

Example:

filter_none
edit

play_arrow

brightness_4

```

using System;
using System.Collections;
class GFG {
public static void Main()
{
Stack myStack = new Stack();
myStack.Push( "1st Element" );
myStack.Push( "2nd Element" );
myStack.Push( "3rd Element" );
myStack.Push( "4th Element" );
myStack.Push( "5th Element" );
myStack.Push( "6th Element" );
Console.Write( "Total number of elements in the Stack are : " );
Console.WriteLine(myStack.Count);
Console.WriteLine( "Element at the top is : " + myStack.Peek());
Console.WriteLine( "Element at the top is : " + myStack.Peek());
Console.Write( "Total number of elements in the Stack are : " );
Console.WriteLine(myStack.Count);
}
}

```

Output:

Total number of elements in the Stack are : 6
Element at the top is : 6th Element
Element at the top is : 6th Element
Total number of elements in the Stack are : 6

Properties

Property	Description
<u>Count</u>	Gets the number of elements contained in the Stack.
<u>IsSynchronized</u>	Gets a value indicating whether access to the Stack is synchronized (thread safe).
<u>SyncRoot</u>	Gets an object that can be used to synchronize access to the Stack.

Example:

filter_none
edit

play_arrow

brightness_4

```

using System;
using System.Collections;
class GFG {
public static void Main()
{
Stack myStack = new Stack();
myStack.Push( "Chandigarh" );
myStack.Push( "Delhi" );
myStack.Push( "Noida" );
myStack.Push( "Himachal" );
myStack.Push( "Punjab" );
myStack.Push( "Jammu" );
Console.Write( "Total number of elements in the Stack are : " );
Console.WriteLine(myStack.Count);
}
}

```

Output:

Total number of elements in the Stack are : 6

Methods

Method	Description
<u>Clear()</u>	Removes all objects from the Stack.
<u>Clone()</u>	Creates a shallow copy of the Stack.
<u>Contains(Object)</u>	Determines whether an element is in the Stack.
<u>CopyTo(Array, Int32)</u>	Copies the Stack to an existing one-dimensional Array, starting at the specified array index.
<u>Equals(Object)</u>	Determines whether the specified object is equal to the current object.
<u>GetEnumerator()</u>	Returns an IEnumerator for the Stack.
<u>GetHashCode()</u>	Serves as the default hash function.
<u>GetType()</u>	Gets the Type of the current instance.
<u>MemberwiseClone()</u>	Creates a shallow copy of the current Object.
<u>Peek()</u>	Returns the object at the top of the Stack without removing it.

<u>Pop()</u>	Removes and returns the object at the top of the Stack.
<u>Push(Object)</u>	Inserts an object at the top of the Stack.
<u>Synchronized(Stack)</u>	Returns a synchronized (thread safe) wrapper for the Stack.
<u>ToArray()</u>	Copies the Stack to a new array.
<u>ToString()</u>	Returns a string that represents the current object.

Example :

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
class GFG {
public static void Main()
{
Stack myStack = new Stack();
myStack.Push( "1st Element" );
myStack.Push( "2nd Element" );
myStack.Push( "3rd Element" );
myStack.Push( "4th Element" );
myStack.Push( "5th Element" );
myStack.Push( "6th Element" );
Console.Write( "Total number of elements in the Stack are : " );
Console.WriteLine(myStack.Count);
myStack.Clear();
Console.Write( "Total number of elements in the Stack are : " );
Console.WriteLine(myStack.Count);
}
}
```

Output:

Total number of elements in the Stack are : 6
Total number of elements in the Stack are : 0

Example :

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
class GFG {
public static void Main()
{
Stack myStack = new Stack();
myStack.Push( "Geeks" );
myStack.Push( "Geeks Classes" );
myStack.Push( "Noida" );
myStack.Push( "Data Structures" );
myStack.Push( "GeeksforGeeks" );
Console.WriteLine(myStack.Contains( "GeeksforGeeks" ));
}
}
```

Output:

True

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/system.collections.stack?view=netframework-4.7.2>

Sahil Bansal

In love with a semicolon because sometimes i miss it so badly)



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.