C# - QUEUE CLASS

It represents a first-in, first out collection of object. It is used when you need a first-in, first-out access of items. When you add an item in the list, it is called **enqueue**, and when you remove an item, it is called **deque**.

Methods and Properties of the Queue Class

The following table lists some of the commonly used **properties** of the **Queue** class –

Sr.No.	Property & Description
1	**Count** Gets the number of elements contained in the Queue.

The following table lists some of the commonly used **methods** of the **Queue** class –

Sr.No.	Method & Description
1	**public virtual void Clear;** Removes all elements from the Queue.
2	**public virtual bool Contains***objectobj***;** Determines whether an element is in the Queue.
3	**public virtual object Dequeue;** Removes and returns the object at the beginning of the Queue.
4	**public virtual void Enqueue***objectobj***;** Adds an object to the end of the Queue.
5	**public virtual object[] ToArray;**

	Copies the Queue to a new array.
6	**public virtual void TrimToSize;** Sets the capacity to the actual number of elements in the Queue.

Example

The following example demonstrates use of Stack –

Live Demo

```
using System;
using System.Collections;

namespace CollectionsApplication {
   class Program {
      static void Main(string[] args) {
         Queue q = new Queue();

         q.Enqueue('A');
         q.Enqueue('M');
         q.Enqueue('G');
         q.Enqueue('W');

         Console.WriteLine("Current queue: ");
         foreach (char c in q) Console.Write(c + " ");

         Console.WriteLine();
         q.Enqueue('V');
         q.Enqueue('H');
         Console.WriteLine("Current queue: ");
         foreach (char c in q) Console.Write(c + " ");

         Console.WriteLine();
         Console.WriteLine("Removing some values ");
         char ch = (char)q.Dequeue();
         Console.WriteLine("The removed value: {0}", ch);
         ch = (char)q.Dequeue();
         Console.WriteLine("The removed value: {0}", ch);

         Console.ReadKey();
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
Current queue:
A M G W
Current queue:
A M G W V H
Removing values
The removed value: A
The removed value: M
```