

## Collection Interview Question and Answers in C#

### What is Collections in C#?

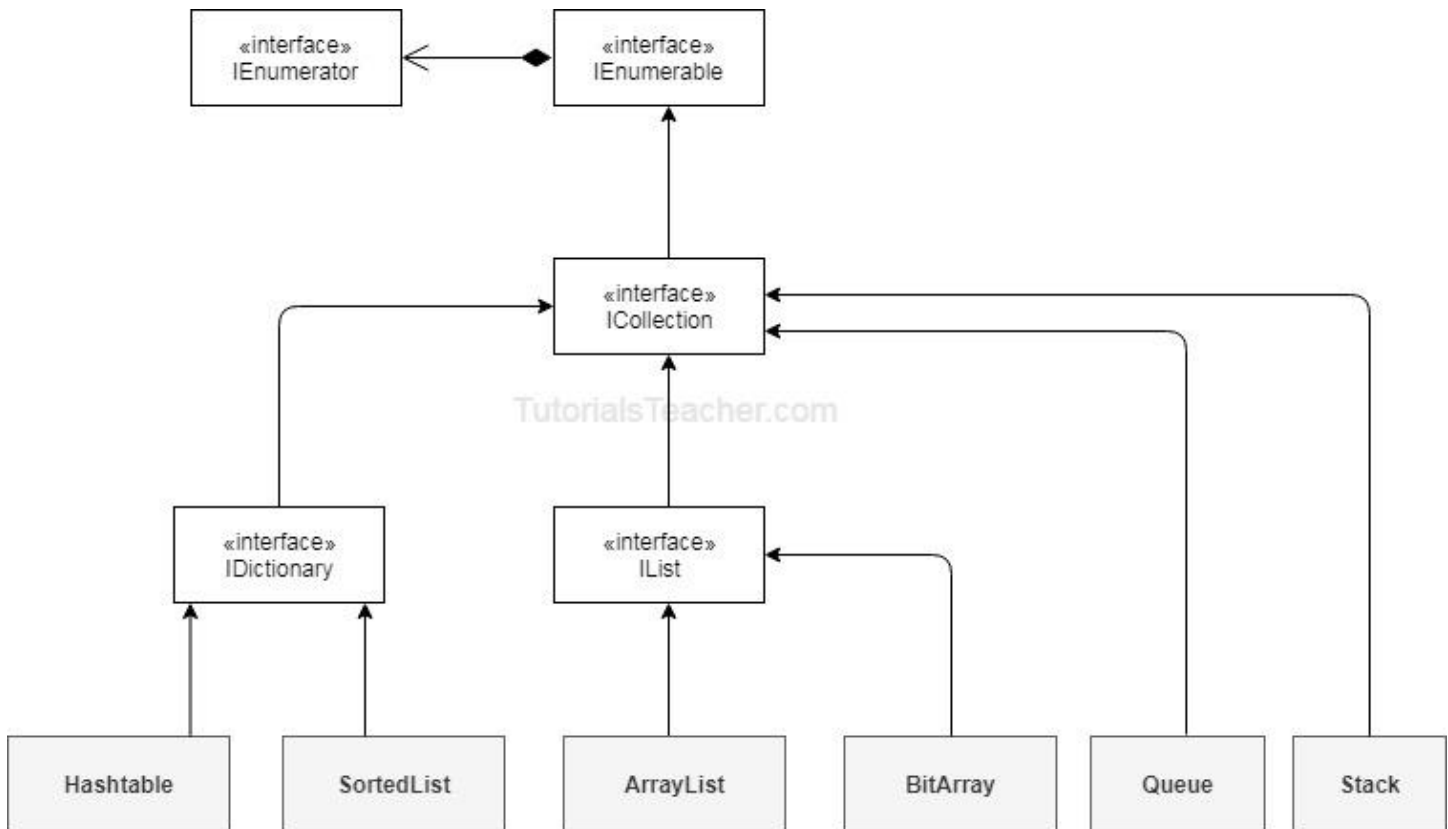
Collections are similar to Arrays, it provides a more flexible way of working with a group of objects.

### What are the types of collection in C#?

There are two type of collection

- Non generic collection
- Generic collection

System.collection



**IEnumerator:** The IEnumerator interface supports a simple iteration over a non-generic collection. It includes methods and property which can be implemented to support easy iteration using foreach loop.

**IEnumerable:** The IEnumerable interface includes GetEnumerator() method which returns an object of IEnumerator.

**ICollection:** The ICollection interface is the base interface for all the collections that defines sizes, enumerators, and synchronization methods for all non-generic collections.

**IList:** The IList interface includes properties and methods to add, insert, remove elements in the collection and also individual element can be accessed by index. The ArrayList and BitArray collections implement IList interface.

**IDictionary:** The IDictionary interface represents a non-generic collection of key/value pairs. The Hashtable and SortedList implement IDictionary interface and so they store key/value pairs.

### What are different non generic collection in C#?

- **ArrayList:** The ArrayList collection is similar to the Arrays data type in C#. The biggest difference is the dynamic nature of the array list collection.
- **Stack:** The stack is a special case collection which represents a last in first out (LIFO) concept
- **Queues:** The Queue is a special case collection which represents a first in first out concept
- **Hashtable:** A hash table is a special collection that is used to store key-value items
- **SortedList:** The SortedList is a collection which stores key-value pairs in the ascending order of key by default.
- **BitArray:** A bit array is an array of data structure which stores bits

### What is ArrayList in C#?

The ArrayList collection is similar to the Arrays data type in C#. The biggest difference is the dynamic nature of the array list collection.

**Syntax:** ArrayList a1 = new ArrayList()

### What is Stack in C#?

The stack is a special case collection which represents a last in first out (LIFO) concept.

**Syntax:** Stack st = new Stack()

## What is Queue in C#?

The Queue is a special case collection which represents a first in first out concept.

**Syntax:** Queue qt = new Queue()

## What is Hashtable in C#?

A hash table is a special collection that is used to store key-value items. So instead of storing just one value like the stack, array list and queue, the hash table stores 2 values. These 2 values form an element of the hash table.

**Syntax:** Hashtable ht = new Hashtable()

## What is SortedList in C#?

**Syntax:** SortedList sortedList1 = new SortedList();

## What is BitArray in C#?

BitArray manages a compact array of bit values, which are represented as Booleans, where true indicates that the bit is on (1) and false indicates the bit is off (0).

**Syntax:** BitArray bits = new BitArray();

```
using System;
using System.Collections;

namespace IntroductionToCollection
{
    class BitArray_Demo
    {
        static void Main()
        {
            bool[] array = new bool[5] {true, false, false, true, true};

            BitArray bits=new BitArray(array);

            foreach (var item in bits)
            {
                Console.WriteLine(item);
            }
        }
    }
}
```

## **What is generic in c#?**

Generics allow you to define a class with placeholders for the type of its fields, methods, parameters, etc. Generics replace these placeholders with some specific type at compile time.

Generics can be applied to the following:

- Interface
- Abstract class
- Class
- Method
- Static method
- Property
- Event
- Delegates
- Operator

## **What are the advantages of Generics**

1. Increases the reusability of the code.
2. Generic are type safe. You get compile time errors if you try to use a different type of data than the one specified in the definition.
3. Generic has a performance advantage because it removes the possibilities of boxing and unboxing.