Difference between array and arraylist in c#.

**Array**

An array is a data structure that contains a number of element of the same type. Array is reference type, so memory on heap must be allocated. This is done by initializing the variable of array using new operator with the type and number of elements inside the array.

*EXAMPLE*

string[] array1=new string[5];

As we have initialized array so array1 is already allocated with 5 empty array element.

After an array is declared and initialized, you can access the array element using an indexer. Arrays only support indexer that have integer parameters.

array1[0]=”Hello”;

array1[1]=”Bye”;

1. Here first important point to note down is that array are fixed size.
2. Second important point is that array are strongly type. It means if you are declaring array of string type then you can not store integer value inside array.
3. Array provide better performance than arraylist because it doesn’t need boxing and unboxing as we store same type of data.
4. Array class is base class for all array in c#. It is defined in the the system namespace.
5. Properties of Array class:

* IsFixedSize
* IsReadOnly
* Length
* Rank

1. Array can not store null.

**ArrayList**

Collection classes are special classes for data storage and retrieval.

*EXAMPLE*

using System.Collection;

ArrayList a1=new ArryList();

a1.add(null);

a1.insert(1,”hi”);

a1.add(3);

a1.add(8.23);

1. Arraylist doesn’t have specific size. Arraylist increase by 4 item. When you initialize arraylist it initially allocate memory for 4 elements. When we add 5th element number of element exceed then its capacity so arraylist will automatically redimension to double of its current size.So, size will increase as 4,8,16,32,64 and so on.
2. Arraylist is non-generic type of collection in c#. It means you can store any type of data in arraylist.
3. Arraylist provide facility of size but it come at cost of performance. The arraylist’s internal array is of object type. So, if you are using value type then each element is boxed and stored on heap and whenever you access arraylist element it is unboxed to value type.
4. Arraylist implement IList interface so, it provide various method that we can use for easy implementation.

* Add() : Add single element to arraylist.
* Insert(): Allow to insert single element at specific position.
* Remove(): Remove single elemet from arraylist.
* RemoveAt(): Remove element from specific postion.

1. To use Arraylist you must add namespace System.Collection.
2. Arrayliat can store null.

**SUMMRY**

|  |  |
| --- | --- |
| **Array** | **ArrayList** |
| Array are strongly type. We can store only same type of data. | Arraylist are non-generic collection type. Arraylist’s internal array is of object type. So, we can store multiple type of data on single arrytlist. |
| Array stores fixed number of elements. | Arraylist are dynamic in term of capacity.  if number of element exceed arraylist will increase to double of its current size |
| Array provide better performance than arraylist. | When we are using large number of arraylist read and write at that time it degrade performance because of boxing and unboxing. |
| Array use static helper class array which belongs to system namespace | Arraylist provide various method which makes implemention of arraylist easy. |
| Array belongs to namespace System | ArrayList belongs to namespace System.Collection |
| Array can not accept null. | Array can accept null. |
| Example:  string[] array1=new string[5];  array1[0]=”Hello”;  array1[1]=”Bye”; | Example:  ArrayList a1=new ArryList();  a1.add(null);  a1.insert(1,”hi”);  a1.add(3);  a1.add(8.23); |