Object Array

Object array is used to store elements of the different types in a single array. In C#, an object reference may point to any derived type instance.

Disadvantages of Object array:

- It makes code more complex.
- It decrease the run-time of the program.

```
// C# program to illustrate the
// concept of object array
using System;

class demo {

    // Main method
    static public void Main()
    {

        // Creating and initializing
        // object array
        object[] arr = new object[6];

        arr[0] = 3.899;
        arr[1] = 3;
        arr[2] = 'l';
        arr[3] = "Lambton";

        // it will display
        // nothing in output
        arr[4] = null;

        // it will show System.Object
        // in output
        arr[5] = new object();

        // Display the element of
        // an object array
        foreach(var item in arr)
        {
            Console.WriteLine(item);
        }
    }
}
```

Output:

```
3.899
```

```
3

1

Lambton



System.Object
```

Dynamic Array

Static arrays have the disadvantage that if you have not used a full array then it will always use the same size as was defined during its declaration. We usually need to have an array that we would not know the values of or how many of them exist. For that, we can use a dynamic array.

A Dynamic Array defines a size of the array at runtime, but then makes room for new elements in the array during execution.

The dynamic array provides dynamic memory allocation, adding, searching, and sorting elements in the array.
Dynamic array overcomes the disadvantage of the static array.
In a static array, the size of the array is fixed but in a dynamic array, the size of the array is defined at run-time.

Declaration and Initialization of Dynamic Array

```
List<data type> name= new List<data type>();
e.g;
List<int> list=new List<int>();
```

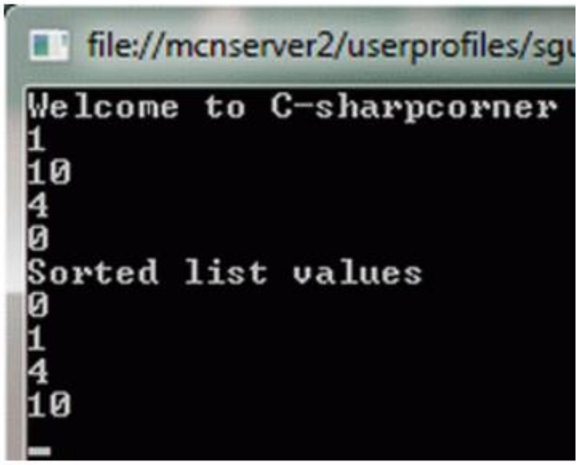
```
// C# program to illustrate the
// concept of dynamic array
using System;
using System.Collections;
using System.Collections.Generic;

class demo1 {

        // Main method
        static public void Main()
        {

                // Creating and initializing
                // the value in a dynamic list
                List<int> myarray = new List<int>();
                myarray.Add(23);
                myarray.Add(1);
                myarray.Add(78);
                myarray.Add(45);
                myarray.Add(543);

                // Display the element of the list
                Console.WriteLine("Elements are: ");
                foreach(int value in myarray)
                {
                    Console.WriteLine(value);
                }

                // Sort the elements of the list
                myarray.Sort();

                // Display the sorted element of list
                Console.WriteLine("Sorted element");
                foreach(int i in myarray)
                {
                    Console.WriteLine(i);
                }
        }
}
```

Output:

```
Elements are:

23

1

78

45

543

Sorted element

1

23

45

78

543
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace DynamicArray
{
  class Program
  {
     static void Main(string[] args)
     {
        Console.WriteLine("Welcome to C-sharpcorner");
        List<int> list=new List<int>();
        list.Add(1);
        list.Add(10);
        list.Add(4);
        list.Add(0);
        int size = list.Count;
        for (int i = 0; i < list.Count; i++)
        Console.WriteLine(list[i]);
        list.Sort();
        Console.WriteLine("Sorted list values");
        for (int i = 0; i < list.Count; i++)
        Console.WriteLine(list[i]);
        Console.ReadKey();
   }
  }
}
```

Output:



```
Welcome to C-sharpcorner
1
10
4
0
Sorted list values
0
1
4
10
_
```