Array in VB.NET

An array is a collection of variables of same data type that share a common name.
An array is also known as sequential list because memory allocation is done sequentially in array. In array all the elements are referenced using the same name but they are distinguished using an index. Thus we have to use the name of array along with index to reference particular element in an array.
Now consider following example:
If you want to store marks of 10 subjects for particular students then you have to declare 10 variables such as Mark1, Mark2… Mark10 which is very lengthy process and takes lots of extra time of the programmer. In this type of situation you can declare an array as shown below:
Dim Marks (9) as Integer
The above statement will create an array of 10 elements which are identifying using Marks (0), Marks (1) … Marks (9).

Types of Array in VB.NET

In Visual Basic there are two types of Array :
(1) Fixed Array: An array in which number of elements is fixed is known as Fixed Size Array.
(2) Dynamic Array: An array in which number of elements is not fixed is known as Dynamic Array.

Single Dimensional Fixed Size Array

An array with only one dimension is known as Single Dimensional array.
Declaring Single Dimensional Array:
While declaring Single Dimensional Array we must have to specify the number of elements that can be hold by an array.
Once the numbers of elements in the array are defined you can not store more elements in that array.
The general syntax for declaring Single Dimensional Array is given below:

Dim ArrayName (Size) as DataType
Here,
(1) ArrayName is any valid variable name.
(2) Size indicates number of elements that can be store in an array.
Example:
Dim mark (10) as Integer

The above statement creates an array of 11 elements starting from an index 0.
By default the index in the array starts with 0.

Multidimensional Fixed Size Array

An array with more then one dimensional is known as Multidimensional array.
Multidimensional array is useful for storing data in tabular form.
In multidimensional array the elements are stored in the form of rows and columns.
The general syntax for declaring two dimensional array is given below:

Dim ArrayName (Row-Size, Column-Size) as DataType
Here,
Row-Size indicates number of rows in the table.
Column-Size indicates number of columns in the table.
Example:
Dim Employee (6, 1) as Integer

Dynamic Array

An array in which numbers of elements are not fixed is known as dynamic array.
While declaring an array if you don't know actual size of an array, it means how many elements can be stored in an array at that time you have to assume the size of an array. Which may leads to following two problems:
(1) If the size of an array is smaller then the number of elements to be stored in an array then you can not store all elements in an array.
(2) If the size of an array is larger then the number of elements to be stored in an array then there is wastage of memory space.
The above problem can be easily solved using the concept of dynamic array.

How to Create Dynamic Array?

While creating dynamic array there is no need to know the actual number of elements to be stored in array. There is no need to specify the size of the array while declaring dynamic array.
The general syntax for declaring dynamic array is given below:

Dim ArrayName () as DataType

If you don't specify the size of an array then it becomes dynamic array in which you can store any number of elements.

Dim Marks () as Integer

Later in the program if you know the exact size of the array then you can specify the size of the array using the ReDim statement as follow:

ReDim Marks (9)

It will set the size of the array to 10 elements.
Remember following thing about ReDim statement:
(1) It must appear inside the procedure. We can not use it outside the procedure like Dim statement.
(2) ReDim statement executes at run time while Dim statement executes at compile time.
(3) Using ReDim statement you cannot change the DataType of the array.

(4) Once you redefine the dynamic array you can change size of the array but you cannot change the dimensional of the array.

Disadvantage of ReDim statement

When you redefine the dynamic array the contents of the array are lost and all the elements are initialized to default value.

Preserving the contents of Dynamic Array

One disadvantage of ReDim statement is that when you redefine the dynamic array the contents of the array are lost and initialize to default value. However if you don't want to lost the contents of the array then you can use the PRESERVE keyword in the ReDim statement as shown below:

ReDim Preserve ArrayName (Size)

The use of the Preserve keyword in the ReDim statement cause the contents of the array to be remained as it is and there is no lost of any data.