Array

An array is a variable that can store data of similar type or homogeneous data. Once the data is stored in the array that data can be accessed through it index. An array can be defined using the following syntax.

<datatype>[]<variable>=new <datatype>[size]

Where data type is the predefined data type supported by C#. Variable is the name of the variable. Size specifies the length of the array.

For example:

```
int[] x=new int[5];
```

x[0]=0,x[1]=0,x[2]=0,x[3]=0,x[4]=0 (default an array initialize with 0)

An array can also be declared using the following syntax:

<datatype>[]<variable>=new <datatype>[]{value1,value2,…valuen};

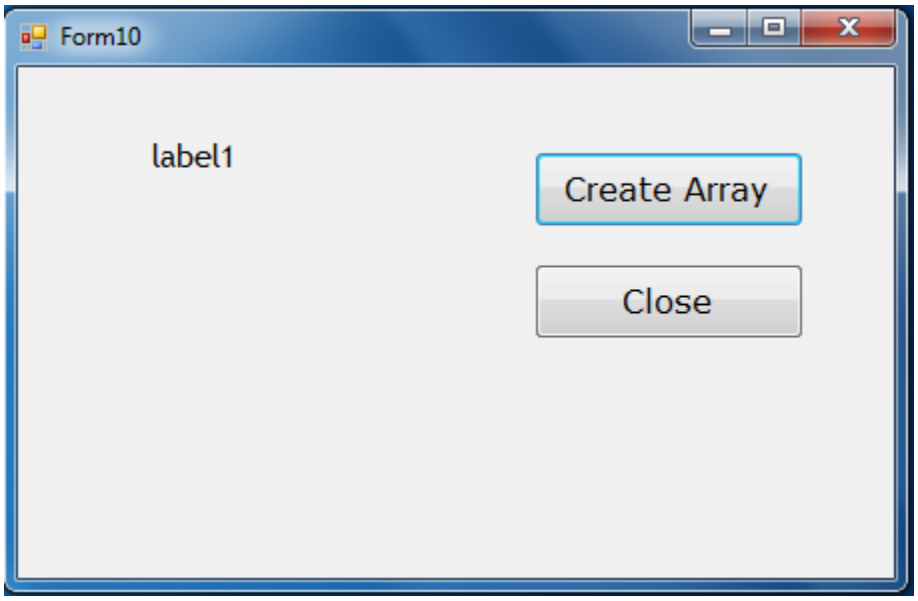
```
int[] x=new int[]{10,20,30};
```

Note:

- The array index starts with zero and the last term index is n-1.
- All numeric type of arrays is initialized to zero by default the string which is initialized to null.
- While defining the array second syntax, the size should not specify the size.
- All array variables are referenced to System.Array class. This class contains the following properties and methods.

Clear()	Clear all the array elements
CopyTo(ArrayVariable,Index)	Copies the current array elements into a different array variable.
Find(value)	Finds the give value in the array.
IndexOf(value)	Returns the index of the given value.
Reverse(arrayvariable)	Arranges the array elements in reverse order.
Sort(arrayvariable)	Arranges the array elements in sorted order that is ascending order for numeric values alphabetics in alphabetic order

Example : WAP to define an Array

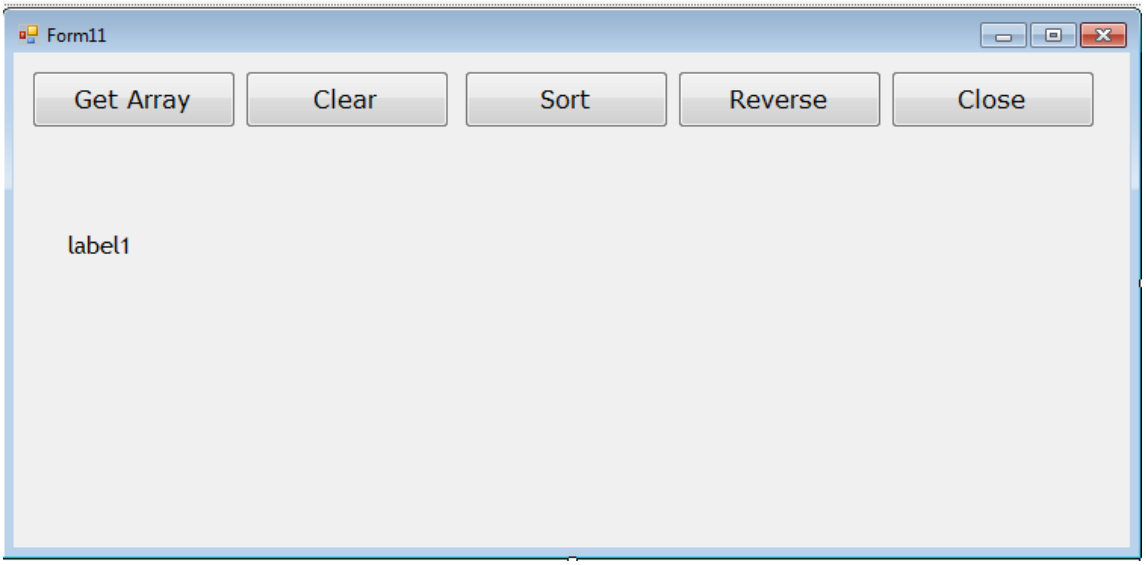



```
//CreateArrary
            int[] x = new int[5];
            x[0] = 10;
            x[1] = 20;
            x[2] = 30;
            x[3] = 40;
            x[4] = 50;

            string s = "";
            for (int i = 0; i < 5; i++)
                s = s + "X[" + i + "]=" + x[i] + "\n";
            label1.Text = s;

            int[] y = new int[] { 10, 20, 30, 40, 50 };
            for(int i=0;i<5;i++)
            {
                s = s + "Y[" + i + "]=" + y[i] + "\n";
            }
            label1.Text = s;
            MessageBox.Show(s, "Arrays", MessageBoxButtons.OK, MessageBoxIcon.Information);
//close
            this.Close();
```

Create an interface to store element into an array, print the array element, sort & reverse the array elements




```
//global
        int[] x = new int[] { 12, 13, 25, 6, 9, 8, 5, 18, 32 };
        string s = "";
//Get Array
            s=s+"The aray elements are...\n";
            for (int i = 0; i < x.Length; i++)
                s = s + "X[" + i + "]=" + x[i]+"\n" ;
            label1.Text = s;
//Clear
            Array.Clear(x,0,x.Length);
            for (int i = 0; i < x.Length; i++)
                s = s + "X[" + i + "]=" + x[i] + "\n";
            label1.Text = s;
  //Sort
            Array.Sort(x);
            for (int i = 0; i < x.Length; i++)
                s = s + "X[" + i + "]=" + x[i] + "\n";
            label1.Text = s;
//Reverse
            Array.Reverse(x);
            for (int i = 0; i < x.Length; i++)
                s = s + "X[" + i + "]=" + x[i] + "\n";
            label1.Text = s;
//close
            this.Close();
```

Double-Dimensional Array

An array can have two dimensional, three dimensional or multi-dimensions.

A two dimensional array can be defined using the following syntax:

<Datatype>[,]<variable>=new <Datatype>[,]

int [,] x=new int[,]

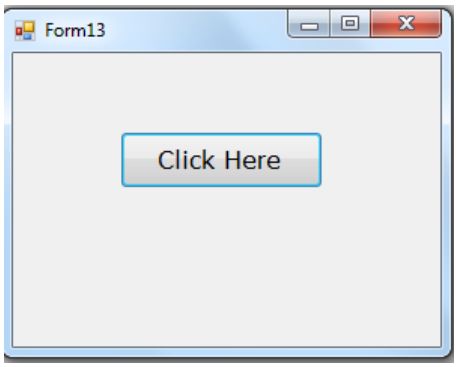
A double dimension array can be defined and initialize using the following syntax:

<datatype>[,] <variable>=new <dataype>[,]{{value1,value2} {value1,value2}};

Forexample,

int[,] x=new int[,]{5,6,7}{3,2,1} {5,8,9}};

Example




```
//click here
            string s = "";
            int[,] x = new[,] { { 5, 6, 7 }, { 3, 4, 6 }, { 9, 8, 4 } };
            MessageBox.Show("Total:" + x.Length);
            MessageBox.Show("Rows:" + x.GetLength(0)+"Column:"+x.GetLength(1));
            for (int i = 0; i < x.GetLength(0); i++)
            {
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    s = s + "X[" + i + "," + j + "]=" + x[i, j] + "";
                }
                s = s + "\n";
            }
            MessageBox.Show(s);
```

Jagged Array:

A Jagged array contains fixed number of rows but uneven number of columns. As the numbers of elements in each row are not uniform, it is called a jagged array.

<datatype>[][] <variable>=new <datatype>[rows][] **//column should not specified**

Ex:

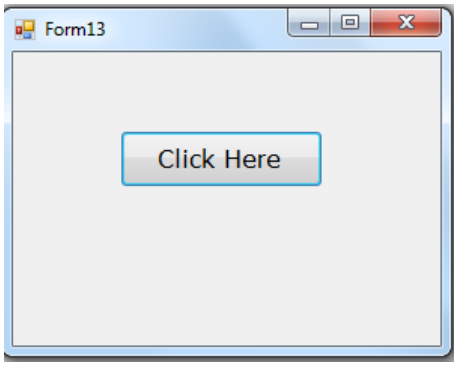
int[][] x=new int[3][];

int [][] y=new int[3][4]; **//Error occurs: Should not specify the column size here 4**

Adv of Jagged Arrays

- They are dynamic in nature.
- They can be used to build the complex applications.
- They can be used to provide solutions for real-time problems.

Example




```
//click here
int[][] x = new int[3][];
            x[0] = new int[] { 3, 4, 5, 6 };
            x[1] = new int[10];
            x[2] = new int[] { 2, 4, 6 };
            string s = "";
            for (int i = 0; i < x.GetLength(0); i++)
            {
                for (int j = 0; j < x[i].Length; j++)
                {
                    s = s + x[i][j] + "";
                }
                s = s + "\n";
            }
            MessageBox.Show(s);
```