

# Arrays

- Array is a collection of similar type of data.
- Array stores multiple values of same type.
- Arrays are reference types.
- They are automatically of a predefined type **System.Array**
- **Syntax:**
- **Datatype [] Array\_name=new Data type[size];**
- **Data type-** specifies what type of data storing into the array. Array type can be value type or reference type.
- **Array\_name**-Indicates name of the array
- **new**-It allocates memory to the array at runtime
- **Size**-Indicates how many elements storing into the array
- ex: `int []a=new int[5];`
- `string []names=new string[3];`

# Arrays

- Assigning values to array

- 1. `Int []a=new int[4]`

- `a[0]=10`

- `a[1]=20`

10	20	30	40
----	----	----	----

- `a[2]=30`

0      1      2      3

- `a[3]=40`

- 2. `Int []a=new int[4]{10,20,30,40}`

- 3. `Int []a={10,20,30,40} //Dynamic allocation`

- Accessing elements

**`a[0];`**

- In arrays index starts from 0

- If there is a mismatch between the declared size and the number of initializes , a compile time error is generated.

# Arrays

- members of `System. Array`
- **`sort()`** -It sorts array elements in ascending order
- **`System.Array.Sort(array name)`**
- **`reverse()`**-It arrange array elements in reverse order
- **`System.Array.Reverse(array name)`**
- **`copy()`** –It copies one Array elements into to another array
- **`System.Array.Copy(source array name, destination array name, count)`**
- **`Length`**- It returns size of the array
- **`Rank`**-It returns dimension of the array

# Multidimensional array

- A multidimensional array stores values in multiple rows and columns.
- An array can have a maximum of 32 dimensions.
- Two types of multidimensional array
  - Rectangular array
  - Jagged array

# Rectangular Array

- A multidimensional array where length of each row is fixed throughout the array.
- Syntax: Data type[,] array\_name=  
new Data type[row\_size, col\_size]

- Creation:

```
int[,] matrix = new int[5,5];
```

- Accessing array elements:

```
matrix[0,1]=9;
```

- Initializing

```
int[,] matrix = new int[,] { {0, 1}, {2, 3},  
    {4, 5} };
```

# Jagged Arrays

- A jagged array is an array of arrays.
- The element within the jagged array itself is an array.
- The arrays may be of different sizes.
- Definition requires specification of size for only the topmost array.
- **Datatype [ ][ ] Array\_name = new Datatype[size][ ] ;**
- Size indicates no of arrays storing into jagged array
- Creation:  

```
int[][] myarr=new int[3][];
```
- Creating arrays in the jagged array,  

```
myarr[1]= new int[3]{0,1,2};
```
- Accessing elements  

```
myarr[0][1]=8;
```
- Accessing the length of row 0 → `myarr[0].Length`
- Initializing
  - ```
int[][] myarr= new int[][] { new int[] {0,1}, new int[] {3,4,5} };
```
  - Or simply 

```
int[][] myarr = { new int[] { 0, 1 }, new int[] { 3, 4, 5 } };
```