# Array

Arrays are powerful data structures in C# that provide a convenient way to store and manipulate collections of data. They are extensively used in various programming scenarios, including algorithms, data processing, and more.

Collection of Data: An array is a data structure that holds a collection of items or elements. These items can be of the same data type (e.g., integers, characters, etc.).

Ordered Sequence: Arrays organize data in an ordered sequence, where each element has a unique position known as an index.

Fixed Size: Arrays have a fixed size, meaning the number of elements they can hold is determined at the time of creation and cannot be changed later.

Zero-Based Indexing: The elements in an array are accessed using indices, which start from 0 and go up to the size of the array minus one.

Random Access: Elements in an array can be accessed directly using their indices. This allows for fast retrieval and modification of individual elements.

Homogeneous Elements: All elements in an array must be of the same data type. For example, an array of integers cannot contain characters or strings.

**Declaration and Initialization:**

* Arrays are declared using square brackets [] after the data type.
* They can be initialized during declaration or separately afterward.
* Arrays can hold elements of any valid data type.

Example:

int[] numbers = new int[5]; // Declaration and initialization

numbers[0] = 1;

numbers[1] = 2;

// ...

**Accessing Elements:**

* Array elements are accessed using zero-based indexing.
* You can read or modify individual elements using their indices.

Example:

int[] numbers = { 1, 2, 3, 4, 5 };

int secondNumber = numbers[1]; // Accessing the second element (index 1)

numbers[3] = 10; // Modifying the fourth element (index 3)

**Length Property:**

* Arrays have a Length property that returns the total number of elements in the array.
* The Length property is read-only and cannot be modified.

Example:

int[] numbers = { 1, 2, 3, 4, 5 };

int arrayLength = numbers.Length; // Returns 5

**Iterating Over Arrays:**

* You can use loops like for, foreach, or while to iterate over the elements of an array.
* foreach loop is commonly used for iterating over array elements.

Example:

int[] numbers = { 1, 2, 3, 4, 5 };

foreach (int num in numbers) {

Console.WriteLine(num);

}

**Multidimensional Arrays:**

* C# supports multidimensional arrays, including rectangular and jagged arrays.
* Rectangular arrays have fixed dimensions, whereas jagged arrays have arrays of arrays.

Example:

int[,] matrix = new int[3, 3]; // Rectangular array

int[][] jaggedArray = new int[3][]; // Jagged array

Jagged array:

A jagged array is a special type of array in programming that is also known as an "array of arrays." Unlike a regular, or rectangular, array where all rows have the same number of columns, a jagged array allows each row to have a different number of columns