```
MVC
                                            Web API
                                                        Design Patterns
                                                                           .NET Core V
                                                                                           Dotnet ~
                                                                                                        Data Bases V
                                                                                                                        Java ∨
                                                                                                                                   C/C++/DSA V
                                                                                                                                                    More... v
Online Training
                                               2D Arrays in C#
 C#.NET Online Training
    Program
                                               Back to: C#.NET Tutorials For Beginners and Professionals

    ✓ ASP.NET Core Training

 Microservices Online Training
    using .NET Core
                                               2D Arrays in C# with Examples
 Microsoft Azure Training
                                               In this article, I am going to discuss the 2D Arrays in C# with Examples. Please read our previous article before proceeding to
Introduction & Environment
                                               this article where we discussed One-Dimensional Arrays in C# with Examples.
Setup
  How Computer Works
                                               What is a Two-Dimensional Array in C#?
 Introduction to Programming
    Languages
                                               The arrays which store the elements in the form of rows and columns are called Two-Dimensional Array in C#. The two-
 How Computer Programs
                                               dimensional array which is also called a multidimensional array is of two types in C#. They are as follows
    Works
 Different Types of
                                                   1. Rectangular Array: The array whose rows and columns are equal is called a rectangular array
    Applications
                                                   2. Jagged Array: The array whose rows and columns are not equal is called a jagged array
 Programming Methodologies
 Algorithm, Pseudocode,
                                               Rectangular 2D Arrays in C#:
    Programs, and Flowcharts
 Introduction to .NET
                                               A two-dimensional array is an array in which each element is referred to by two indexes. Element in the 2D array is stored in
    Framework
                                               the matrix form. The first index shows the row of the matrix and the second index shows the column of the matrix.
 .NET Framework Architecture
    and Components
                                               Example: int[,] matrix = new int[3,3];
 ✓ Introduction to C#
    Programming Language
                                               2D Array in Memory Representation is shown below. To access elements in the zeroth index we need to specify two indexes
 How to Download and Install
                                               matrix[0][0].
    Visual Studio on Windows
 Creating First Console
                                                                                  matrix[0][2]
                                                  matrix[0][0]
                                                                  matrix[0][1]
    Application using Visual
                                                  matrix[1][0]
                                                                  Matrix[1][1]
                                                                                  Matrix[1][2]
    Studio
                                                                 Matrix[2][1]
                                                  Matrix[2][0]
                                                                                  Matrix[2][2]
 .NET Developer Roadmap for
    2025
                                                           2D-Matrix
 Coding Standard Best
    Practices
                                               How 2D Array is Created and Accessed in C#?
C#.NET Basics
                                               The method for creating a rectangular two-dimensional array is as follows:
 int[,] A = new int[3,4];
    Program
                                               If we created it like this, then the 2D array is created with 3 rows and 4 columns where the name of the array is A. For a better
 Methods and Properties of
                                               understanding, please have a look at the below diagram.
    Console Class in C#
 Literals in C#
                                                                 2
 Type Casting in C#
                                                        0
                                                            1
                                                                     3
 ✓ Variables in C#
                                                   0
 Operators in C#
 Control Flow Statements in
    C#

✓ If-Else Statements in C#

 Switch Statements in C#
                                               Here, j represents the row number and i represents the column number. As we created the array with sizes 3 and 4, here you
 ✓ Loops in C#
                                               can see that the array is created with 3 rows and 4 columns. We can access any element with the row and column number as

    While Loop in C#

                                               follows:
 Do While Loop in C#
                                               Console.WriteLine(A[1,2]);
 For Loop in C#
                                               This means the 2nd row and 3rd column as Array Indexed are 0-based indexes.
 Break Statement in C#
 Continue Statement in C#
                                               Note: Indexing starts from 0 onwards in the array. So, we have started the row and column from 0. This is how we can access
 Goto Statement in C#
                                               any location. Addressing of 2d array is mapped the same as single dimension array.
 Functions in C#
 User-Defined Functions in C#
 Call By Value and Call By
    Reference in C#
 Recursion in C#
 User Input and Output in C#
 Command Line Arguments in
    C#
 String in C#
                                               Memory Locations are allocated continuously side by side. So basically, it will create a single dimension array of size 12,
 Static Keyword in C#
                                               where the first four locations are used as the first rows, the second four locations are used as the second row and the rest four
 Static vs Non-Static Members
                                               locations are used as the third row. So, in a computer the memory is represented as follows:
    in C#
 Const and Read-Only in C#
 Properties in C#
 Why we Should Override
                                                                                              row 3
                                                                           row 2
                                                          row 1
    ToString Method in C#
 Override Equals Method in C#
 Difference Between
                                               But the compiler will allow us to access this single-dimension array as a 2D array. Next, let us see how to create and initialize
    Convert.ToString and
                                               a 2D array.
    ToString Method in c#
 Checked and Unchecked Key
                                               Initializing a 2Dimensional Array in C#:
    words in C#
 Stack and Heap Memory in
                                               Let us understand how to initialize a 2D Array with an example. Please have a look at the following statement which shows the
    .NET
                                               declaration and initialization of a 2D Array.
 Boxing and Unboxing in C#
                                               int[,] A = \{\{2, 5, 9\}, \{6, 9, 15\}\};
                                               This is the declaration + initialization of a 2Dimensinal array in C#. Here 2,5,9 is the 1st row and 6,9,15 is the 2nd row. This is
OOPs in C#
                                               how they will be filled and we can access any element with the help of two indexes that is row number and column number.
 Object Oriented Programming
                                               Now, the other way of initializing it is,
    (OOPs) in C#
                                               int[,] A = new int[2,3]
 Class and Objects in C#
 Constructors in C#
 Types of Constructors in C#
                                                   {2, 5, 9},{6, 9, 15}
 Why We Need Constructors
                                               };
    in C#
                                               Later part of this article, we will see how to initialize the 2D Array dynamically in C#. The following image shows the syntax of
 Static vs Non-Static
                                               how we can initialize a 2D Rectangle Array in C#.
    Constructors in C#
 Private Constructors in C#
                                                 Syntax:
 Destructors in C#
                                                 <type>[,] <name> = new <type> [rows, cols];
 Garbage Collection in .NET
    Framework
                                                 Example:
 Differences Between Finalize
                                                 int [,] arr = new int [3,4]
    and Dispose in C#
                                                 Or
 Access Specifiers in C#
                                                 int [,] arr;
 Encapsulation in C#
                                                 arr = new int [2,3];
 Abstraction in C#
 Inheritance in C#
                                                 Or
 Types of Inheritance in C#
                                                 int [,] arr = {list of values};
 How to use Inheritance in
    Application Development
                                               Next, let us see how to access the elements of the 2-D array.
 C#
                                               Accessing the Elements of the 2D array in C#:
 Generalization and
    Specialization in C#
                                               For accessing all the elements of the rectangle 2D Array in C#, we require a nested for loop, one for loop for accessing the
 Abstract Class and Abstract
                                               rows, and another for loop for accessing the columns. So, by using a nested for loop we can access 2Dimensional Array
    Methods in C#
                                               Elements. For a better understanding, please have a look at the below image.
 Abstract Class and Abstract
    Methods Interview Questions
    in C#
                                                   //2D Array with 3 Rows and 3 Columns

    ✓ How to Use Abstract Classes

                                                   int[,] A = new int[2, 3]
    and Methods in C#
                                                    {
    Application
                                                         {2, 5, 9}, {6, 9, 15}
 ✓ Interface in C#
                                                   };
 Interface Interview Questions
    and Answers in C#
                                                   //Accessing Array Elements using nested for loop
 Interface Realtime Examples
                                                   for (int i = 0; i < 2; i++) //Accessing the Rows
    in C#
 Multiple Inheritance in C#
                                                         for (int j = 0; j < 3; j++) //Accessing the Columns
 Multiple Inheritance Realtime
    Example in C#
 Polymorphism in C#
                                                                Console.Write(A[i, j]); //Accessing the Array Elements
 Method Overloading in C#
 Operator Overloading in C#
                                                   }
 Method Overriding in C#
 Method Hiding in C#
                                               Example to Understand 2D Array in C#:
 Partial Class and Partial
    Methods in C#
                                               Let us see an example for a better understanding of the rectangular 2D array in C#. In the below example, we are creating a
 Sealed Class and Sealed
                                               two-dimensional integer array with 4 Rows and 5 Columns. Then we are printing the values of the 2D Array using a for each
    Methods in C#
                                               loop to see what default it stores. Then using nested for loop, we are assigning the values to the 2D Array as well as Printing
 Extension Methods in C#
                                               the values of the 2D Array. In the below example, we are using the Array class GetLength method, when we pass 0, it will
 Static Class in C#
                                               return the size of the Rows and when we pass 1, it will return the size of the columns.

    ✓ Variable Reference and

    Instance of a Class in C#
                                                  using System;
                                                  namespace TwoDimensionalArayDemo
OOPs Real-Time Examples
 Real-time Examples of
                                                       class Program
    Encapsulation Principle in C#
                                                           static void Main(string[] args)
 Real-Time Examples of
    Abstraction Principle in C#
 Real-Time Examples of
                                                                int[,] RectangleArray = new int[4, 5];
                                                                int a = 0;
    Inheritance Principle in C#
 Real-Time Examples of
    Polymorphism Principle in C#
 Real-Time Examples of
                                                                foreach (int i in RectangleArray)
    Interface in C#
 Real-Time Examples of
                                                                     Console.Write(i + " ");
    Abstract Class in C#
                                                                Console.WriteLine("\n");
Exception Handling
 Exception Handling in C#
 Multiple Catch Blocks in C#
 Finally Block in C#
                                                                for (int i = 0; i < RectangleArray.GetLength(0); i++)</pre>
 How to Create Custom
                                                                     for (int j = 0; j < RectangleArray.GetLength(1); j++)</pre>
    Exceptions in C#
 Inner Exception in C#
                                                                          a += 5;
                                                                          RectangleArray[i, j] = a;
 Exception Handling Abuse in
    C#
Events, Delegates and Lambda
Expression in C#
 Course Structure of Events.
                                                                for (int i = 0; i < RectangleArray.GetLength(0); i++)</pre>
    Delegates and Lambda
    Expression
                                                                     for (int j = 0; j < RectangleArray.GetLength(1); j++)</pre>
 Roles of Events, Delegates
                                                                          Console.Write(RectangleArray[i, j] + " ");
    and Event Handler in C#
 Delegates in C#
                                                                Console.ReadKey();
 Multicast Delegates in C#
 Delegates Real-Time Example
    in C#
 Generic Delegates in C#
 Anonymous Method in C#
                                               Output:
 Lambda Expressions in C#
                                                000000000000000000000
 Events in C# with Examples
                                                  10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
Multi-Threading
 Multithreading in C#
                                               In the above example, we assigned the two-dimensional array elements using nested for loop. It is also possible that we can
 Thread class in C#
                                               assign the values to a two-dimensional array in C# at the time of its declaration.
 How to Pass Data to Thread
    Function in Type Safe Manner
                                               2D Array Declaration and Initialization at the Same Statement:
    in C#
 How to Retrieve Data from a
                                               In the below example, we are assigning values to the two-dimensional array at the time of its declaration. Here, we do not
    Thread Function in C#
                                               need to specify the size as based on the argument values it will automatically choose the size. Here, it will create the array
 Join Method and IsAlive
                                               with the size of 3 Rows and 4 Columns. Then, we Printed the Array Elements using both ForEach Loop and Nested for Loop.
    Property of Thread Class in
                                               In the nested for loop, the first for loop, loop variable I will point to the Rows, and in the second for loop, the loop variable I will
                                               point to the columns of the 2D Array.
 Thread Synchronization in C#
 Lock in C#
                                                  using System;

    ✓ Monitor Class in C#

                                                  namespace TwoDimensionalArayDemo
 Mutex Class in C#
                                                      class Program
 Semaphore Class in C#
 SemaphoreSlim Class in C#
                                                           static void Main(string[] args)
 Deadlock in C#
 Performance Testing of a
    Multithreaded Application
                                                                int[,] NumbersArray = {{11,12,13,14},
 Thread Pool in C#
                                                                                  {21,22,23,24},
 Foreground and Background
                                                                                  {31,32,33,34}};
    Threads in C#
 AutoResetEvent and
                                                                Console.WriteLine("Printing Array Elements using ForEach loop");
    ManualResetEvent in C#
                                                                foreach (int i in NumbersArray)

    ▼ Thread Life Cycle in C#

                                                                     Console.Write(i + " ");

    ▼ Threads Priorities in C#

 How to Terminate a Thread in
    C#
                                                                //Printing Array Elements using nested for each
                                                                Console.WriteLine("\n\nPrinting Array Elements using Nested For Loop");
 Inter Thread Communication
                                                                for (int i = 0; i < NumbersArray.GetLength(0); i++)</pre>
    in C#
  How to Debug a Multi-
                                                                     for (int j = 0; j < NumbersArray.GetLength(1); j++)</pre>
    threaded Application in C#
                                                                          Console.Write(NumbersArray[i, j] + " ");
Collections in C#
 Arrays in C#
 2D Arrays in C#
                                                                Console.ReadKey();
 Advantages and
    Disadvantages of Arrays in C#
 Collections in C#
 ArrayList in C#
                                               Output:
 ✓ Hashtable in C#
                                                Printing Array Elements using ForEach loop
 Non-Generic Stack in C#
                                                11 12 13 14 21 22 23 24 31 32 33 34
 Non-Generic Queue in C#
                                               Printing Array Elements using Nested For Loop
 Non-Generic SortedList in C#
                                                 .1 12 13 14 21 22 23 24 31 32 33 34
 Advantages and
    Disadvantages of Non-
                                               Program to Add 2 Matrices using C#:
    Generic Collection in C#
 Generic Collections in C#
                                                  using System;
                                                  namespace TwoDimensionalArayDemo
 Generics in C#
 Generic Constraints in C#
                                                       class Program
 Generic List Collection in C#
                                                           static void Main(string[] args)
 How to Sort a List of
    Complex Type in C#
                                                                Console.WriteLine("Enter rows and column of Matrices: ");
 Comparison Delegate in C#
                                                                int Rows = Convert.ToInt32(Console.ReadLine());
                                                                int Columns = Convert.ToInt32(Console.ReadLine());
 Dictionary Collection Class in
    C#
                                                                //Create 3 2D Arrays with the above size
 Conversion Between Array
                                                                int[,] Matrix1 = new int[Rows, Columns];
                                                                int[,] Matrix2 = new int[Rows, Columns];
    List and Dictionary in C#
                                                                int[,] ResultMatrix = new int[Rows, Columns];
 List vs Dictionary in C#
 Generic Stack Collection
                                                                //Enter Matrix 1 Elements
    Class in C#
                                                                Console.WriteLine("\nEnter Elements of 1st Matrix:");
                                                                for (int i = 0; i < Rows; i++)
 Generic Queue Collection
    Class in C#
                                                                     for (int j = 0; j < Columns; j++)</pre>
 Foreach Loop in C#
                                                                          Matrix1[i, j] = Convert.ToInt32(Console.ReadLine());
 Generic HashSet Collection
    Class in C#
 Generic SortedList Collection
                                                                //Enter Matrix 2 Elements
    Class in C#
                                                                Console.WriteLine("\nEnter Elements of 2nd Matrix:");
 for (int i = 0; i < Rows; i++)</pre>
    Class in C#
                                                                     for (int j = 0; j < Columns; j++)</pre>
 Generic SortedDictionary
    Collection Class in C#
                                                                          Matrix2[i, j] = Convert.ToInt32(Console.ReadLine());
 Generic LinkedList Collection
    Class in C#
 Concurrent Collection in C#
 ConcurrentDictionary
                                                                Console.WriteLine("\nSum of Both the Matrics:");
    Collection Class in C#
                                                                for (int i = 0; i < 2; i++)
 ConcurrentQueue Collection
                                                                     for (int j = 0; j < 3; j++)
    Class in C#
 ConcurrentStack Collection
                                                                          ResultMatrix[i,j] = Matrix1[i,j] + Matrix2[i,j];
    Class in C#
                                                                          Console.Write($"{ResultMatrix[i, j]} ");
 ConcurrentBag Collection
    Class in C#
                                                                     Console.WriteLine();
 BlockingCollection in C#
                                                                Console.ReadKey();
File Handling
 File Handling in C#
 FileStream Class in C#
 StreamReader and
                                               Output:
    StreamWriter in C#
  File Class in C#
                                                Enter rows and column of Matrices:
 TextWriter and TextReader in
    C#
 BinaryWriter and
                                                Enter Elements of 1st Matrix:
    BinaryReader in C#
 StringWriter and
    StringReader in C#
 FileInfo Class in C#
 DirectoryInfo Class in C#
 Export and Import Excel Data
                                                Enter Elements of 2nd Matrix:
    in C#
Asynchronous Programming
 Introduction to Concurrency
 Async and Await in C#
 Task in C#
 How to Return a Value from
                                                Sum of Both the Matrics:
                                                 7 9
    Task in C#
                                                11 13 15
 How to Execute Multiple
    Tasks in C#
                                               Jagged Array in C#:
 How to Limit Number of
    Concurrent Tasks in C#
                                               These are also two-dimensional arrays that will also store the data in the forms of rows and columns. But here in the jagged
 How to Cancel a Task in C#
                                               array, the column size will differ from row to row. That means if the first row contains 5 columns, then the second row may
    using Cancellation Token
                                               contain 4 columns while the third row may contain 10 columns. So, the point that you need to remember is if the column size
  How to Create Synchronous
                                               varies from row to row then it is a jagged array. If the column size remains the same for all the rows, then it is a rectangular
    Method using Task in C#
                                               two-dimensional array.
 Retry Pattern in C#
 Only One Pattern in C#
                                               The jagged array in C# is also called the array of arrays. This is because in the case of the jagged array each row is a single-
 How to Control the Result of
                                               dimensional array. So, a combination of multiple single-dimensional arrays with different column sizes forms a jagged array in
    a Task in C#
                                               C#.

    ▼ Task-Based Asynchronous

    Programming in C#
                                               Syntax: <type> [][] <name> = new <type> [rows][];
 Chaining Tasks by Using
                                               Example:
    Continuation Tasks
                                               int [][] arr = new int[3][];
 How to Attached Child Tasks
    to a Parent Task in C#
                                               //Or
 ✓ ValueTask in C#
                                               int [][] arr = {list of values};
 How to Cancel a Non-
    Cancellable Task in C#
                                               To declare a jagged array in C#, at the time of its declaration, you only need to specify the number of rows that you want in the
 Asynchronous Streams in C#
                                               array. For example
 How to Cancel Asynchronous
                                               int [][] arr = new int[4][];
    Stream in C#
                                               In the above array declaration, we are specifying that we want four rows in the array. Once you specify the number of rows
Parallel Programming
                                               that you want in the array, then you need to initialize each row with the number of columns by using a single-dimensional array
 Task Parallel Library in C#
                                               as shown below.
 Parallel For in C#
                                               arr[0] = new int[5]; // we want five columns in the first row
 Parallel Foreach Loop in C#
                                               arr[1] = new int[6]; // we want six columns in the first row
 Parallel Invoke in C#
                                               arr[2] = new int[4]; // we want four columns in the first row
 Maximum Degree of
                                               arr[3] = new int[5]; // we want five columns in the first row
    Parallelism in C#
 How to Cancel Parallel
                                               Example to Understand Jagged Array in C#:
    Operations in C#
 Atomic Methods Thread
                                               Jagged Arrays in C# are nothing but the combination of multiple 1D Arrays. In the below example, we created one jagged
    Safety and Race Conditions in
                                               array with 4 Rows and then we initialize each row of the Jagged Array with different 1-Dimensional Arrays. Then we print the
    C#
                                               default values of jagged array using the nested for loop and then we initialize the jagged array using the nested for loop. And
 Interlocked vs Lock in C#
                                               finally, once we initialized the jagged array, then we print the elements of a jagged array using a for loop and for each loop.
 Parallel LINQ in C#
 Multithreading vs
                                                 using System;
    Asynchronous Programming
                                                  namespace TwoDimensionalArayDemo
    vs Parallel Programming in C#
                                                      class Program
AutoMapper
                                                           static void Main(string[] args)
 AutoMapper in C#
 AutoMapper Complex
    Mapping in C#
                                                                int[][] arr = new int[4][];
  How to Map Complex Type to
                                                                //Initializing each row with different column size
    Primitive Type using
    AutoMapper in C#
                                                                arr[0] = new int[5];
 AutoMapper Reverse Mapping
                                                                arr[1] = new int[6];
                                                                arr[2] = new int[4];
    in C#
                                                                arr[3] = new int[5];
 AutoMapper Conditional
    Mapping in C#
 AutoMapper Ignore Method in
                                                                //GetLength(0): Returns the Size of the Rows (4)
 Fixed and Dynamic Values in
                                                                Console.WriteLine("Printing the Default Values of Jagged Array:");
                                                                for (int i = 0; i < arr.GetLength(0); i++)</pre>
    Destination Property in
    AutoMapper
                                                                     //arr[i].Length: Returns the Length of Each Row
                                                                     for (int j = 0; j < arr[i].Length; j++)</pre>
Optional Parameter, Indexers
and Enums
                                                                          Console.Write(arr[i][j] + " ");
 How to make Optional
    Parameters in C#
 ✓ Indexers in C#
                                                                for (int i = 0; i < arr.GetLength(0); i++)</pre>
 Indexers Real-Time Example
    in C#
                                                                     int num = 10;
 Enums in C#
                                                                     for (int j = 0; j < arr[i].Length; j++)</pre>
.NET Framework Architecture
                                                                          num++;
                                                                          arr[i][j] = num;
 DOT NET Framework
 Common Language Runtime
    in .NET Framework
                                                                //Printing the values of Jagged array by using foreach loop within for loop
 .NET Program Execution
                                                                Console.WriteLine("\n\nPrinting the Values of Jagged Array:");
    Process
                                                                for (int i = 0; i < arr.GetLength(0); i++)</pre>
 Intermediate Language
    (ILDASM & ILASM) Code in
                                                                     foreach (int x in arr[i])
                                                                          Console.Write(x + " ");
 Common Type System in
    .NET Framework
 Common Language
    Specification in .NET
    Framework
 Managed and Unmanaged
    Code in .NET Framework
                                                                Console.ReadKey();
 Assembly DLL EXE in .NET
    Framework
 App Domain in .NET
    Framework
                                               Output:
 Strong and Weak Assemblies
    in .NET Framework
 How to Install an Assembly
    into GAC in .NET Framework
 DLL Hell Problem and
    Solution in .NET Framework
Var, Dynamic and Reflection
 Reflection in C#
 Dynamic Type in C#

    ✓ Var Keyword in C#

✓ Var vs Dynamic in C#

 Dynamic vs Reflection in C#

✓ Volatile Keyword in C#

 Ref vs Out in C#
 Named Parameters in C#
C# 7.X new Features
 C# 7 New Features
 Enhancement in Out Variables
    in C# 7
 Pattern Matching in C#
 Digit Separators in C# 7
 ▼ Tuples in C# 7
                                               In the next article, I am going to discuss the Advantages and Disadvantages of Arrays in C# with Examples. Here, in this
 Splitting Tuples in C# 7
                                               article, I try to explain the 2D Arrays in C# with Examples. I hope this article will help you with your needs. I would like to have
 Local Functions in C# 7
                                               your feedback. Please post your feedback, question, or comments about this 2Dimensional Arrays in C# with Examples
 Ref Returns and Ref Locals in
                                               article.
    C# 7
 Generalized Async Return
    Types in C# 7
 Expression Bodied Members
                                                                     Dot Net Tutorials
    in C#
                                                                     About the Author: Pranaya Rout

    ▼ Thrown Expression in C#

 Async Main in C#
                                                     Pranaya Rout has published more than 3,000 articles in his 11-year career. Pranaya Rout has very good experience with
C# 8 New Features
                                                     Microsoft Technologies, Including C#, VB, ASP.NET MVC, ASP.NET Web API, EF, EF Core, ADO.NET, LINQ, SQL
                                                     Server, MYSQL, Oracle, ASP.NET Core, Cloud Computing, Microservices, Design Patterns and still learning new
 C# 8 New Features
                                                     technologies.
 ReadOnly Structs in C#
 Default Interface Methods in
                                                    f in X 🖸 🕲 🦪
    C#
 Pattern Matching in C#
 Using Declarations in C#
 Static Local Functions in C#
 Disposable Ref Structs in C#
                                                   Registration Open - Full-Stack .NET with Angular & ASP.NET
 Nullable Reference Types in
                                                   Core
    C#8
 Asynchronous Streams in C#
                                                     New Batch Starts: 15th September, 2025
 Asynchronous Disposable in
                                                     Session Time: 8:30 PM – 10:00 PM IST
 ✓ Indices and Ranges in C#
 Null-Coalescing Assignment
                                                      Advance your career with our expert-led, hands-on live training program. Get complete course details, the
    Operator in C#
                                                                       syllabus, and Zoom credentials for demo sessions via the links below.
 Unmanaged Constructed
                                                      View Course Details & Get Demo Credentials
    Types in C#
 Stackalloc in in C#
                                                   Registration Form
Most Popular C# Books
                                                   Join Telegram Group
 Most Recommended C#
                                                   Contact: +91 70218 01173 (Call / WhatsApp)
    Books
 Most Recommended Data
    Structure and Algorithms
    Books using C#
                                                  Previous Lesson
                                                                                                                                                 Next Lesson
                                                  Arrays in C#
                                                                                                                 Advantages and Disadvantages of Arrays in C#
                                               2 thoughts on "2D Arrays in C#"
                                                        INDHUMATHI
                                                        MARCH 23, 2024 AT 9:54 AM
                                               typing error under the example
                                               //arr.GetLength(0): Returns the size of the Column
                                               instead 0 it should be 1
                                                                                                                                                          Reply
                                                        INDHUMATHI
                                                        MARCH 23, 2024 AT 10:04 AM
                                               arr[0] = new int[5]; // we want five columns in the first row
                                               arr[1] = new int[6]; // we want six columns in the first row
                                               arr[2] = new int[4]; // we want four columns in the first row
                                               arr[3] = new int[5]; // we want five columns in the first row
                                               Need to update the comment, the "first row, second row, third row" instead repeating "first row"
                                                                                                                                                          Reply
                                               Leave a Reply
                                               Your email address will not be published. Required fields are marked *
                                               Comment *
                                                 Name*
                                                                                        Email*
                                                                                                                               Website
                                                 Post Comment
        About Us Privacy Policy Contact ADO.NET Tutorial Angular Tutorials ASP.NET Core Blazor Tuturials ASP.NET Core Tutorials ASP.NET MVC Tutorials
       ASP.NET Web API Tutorials C Tutorials C#.NET Programs Tutorials C#.NET Tutorials Cloud Computing Tutorials Data Structures and Algorithms Tutorials
 Design Patterns Tutorials DotNet Interview Questions and Answers Core Java Tutorials Entity Framework Tutorials JavaScript Tutorials LINQ Tutorials Python Tutorials
```

SOLID Principles Tutorials SQL Server Tutorials Trading Tutorials JDBC Tutorials Java Servlets Tutorials Java Struts Tutorials C++ Tutorials JSP Tutorials

MySQL Tutorials Oracle Tutorials ASP.NET Core Web API Tutorials HTML Tutorials

© Dot Net Tutorials | Website Design by Sunrise Pixel

