**C#.NET 8.0**

**Harsha Vardhan**

.NET Expert

**C#.NET 8.0**

**.NET Fundamentals........................................................................................................................ 26** Introduction to .NET.............................................................................................................26 What is .NET? .......................................................................................................................26 Parts of .NET.........................................................................................................................26 History of .NET......................................................................................................................27 History of .NET......................................................................................................................27 Versions of .NET ...................................................................................................................29 CIL and CLR ...........................................................................................................................30 Execution Model of .NET Programs .....................................................................................30 CIL (Common Intermediate Language) or MSIL ...................................................................31 CLR (Common Language Runtime).......................................................................................31 Sub Components of CLR.......................................................................................................32 .NET Framework Architecture..............................................................................................34 What is .NET Framework......................................................................................................34 System Requirements of .NET Framework: .........................................................................34 .NET Framework Architecture..............................................................................................35 CLS and CTS ..........................................................................................................................36 CLS (Common Language Specification)................................................................................36 CTS (Common Type System) ................................................................................................36 FCL and BCL ..........................................................................................................................37 FCL (Framework Class Library) .............................................................................................37 Introduction to Visual Studio ...............................................................................................38 What is Visual Studio............................................................................................................38 Versions of Visual Studio......................................................................................................38 System Requirements of Visual Studio 2019: ......................................................................39 Project ..................................................................................................................................40 Solution ................................................................................................................................40 File Types of .NET .................................................................................................................40 Folder Structure of .NET Project in Visual Studio ................................................................41

D. Harsha Vardhan (.NET Expert) P a g e 2 | 1583

**C#.NET 8.0**

Visual Studio - Installation....................................................................................................42 Steps to install “Visual Studio 2019”:...................................................................................42 **C#.NET – Language Fundamentals..........................................................................................55** Introduction to C#.NET.........................................................................................................55 What is C#.NET:....................................................................................................................55 Versions of C#.NET ...............................................................................................................56 Tokens of C#.NET..................................................................................................................56 Naming Conventions for Identifiers.....................................................................................57 Identifier Naming Rules........................................................................................................57 Console Applications - Introduction.....................................................................................58 What is Console Application? ..............................................................................................58 Rules of Console Application................................................................................................58 Console Application - Example.............................................................................................59 The “System.Console” class .................................................................................................60 1. System.Console.Write()....................................................................................................60 2. System.Console.WriteLine().............................................................................................61 3. System.Console.ReadKey()...............................................................................................61 4. System.Console.ReadLine()..............................................................................................61 5. System.Console.Clear() ....................................................................................................61 System.Console.WriteLine – Example..................................................................................62 System.Console.ReadKey – Example ...................................................................................63 System.Console.Clear - Example..........................................................................................65 Variables...............................................................................................................................68 Variables - Example..............................................................................................................69 Data Types............................................................................................................................71 Types of Data types..............................................................................................................71 Numerical Data Types..........................................................................................................71 Default Numerical data types ..............................................................................................72 Non-Numerical Data Types..................................................................................................73 Numerical Data Types - Example .........................................................................................73 MinValue and MaxValue......................................................................................................76

D. Harsha Vardhan (.NET Expert) P a g e 3 | 1583

**C#.NET 8.0**

MinValue and MaxValue - Example .....................................................................................76 char - Example......................................................................................................................78 String ....................................................................................................................................79 String - Example....................................................................................................................80 Bool - Example......................................................................................................................81 Operators .............................................................................................................................83 Arithmetical Operators ........................................................................................................83 Arithmetical Operators - Example........................................................................................84 Assignment Operators..........................................................................................................86 Assignment Operators - Example.........................................................................................86 Increment / Decrement Operators......................................................................................88 Increment / Decrement Operators - Example .....................................................................88 Relational Operators............................................................................................................90 Relational Operators - Example ...........................................................................................90 Logical Operators .................................................................................................................92 Logical Operators - Example.................................................................................................93 Concatenation Operator ......................................................................................................94 Concatenation Operator - Example .....................................................................................95 Conditional Operator ...........................................................................................................96 Conditional Operator - Example...........................................................................................96 Control Statements..............................................................................................................98 Types of control statements: ...............................................................................................98 If............................................................................................................................................99 If - Example.........................................................................................................................101 If – Else - Example...............................................................................................................102 Else – If - Example...............................................................................................................104 Nested If - Example ............................................................................................................105 Switch – Case......................................................................................................................107 Switch – Case - Example.....................................................................................................107 While ..................................................................................................................................109 While - Example..................................................................................................................109

D. Harsha Vardhan (.NET Expert) P a g e 4 | 1583

**C#.NET 8.0**

Do – While..........................................................................................................................111 Do – While - Example .........................................................................................................111 for.......................................................................................................................................113 for - Example ......................................................................................................................113 Break...................................................................................................................................115 Break - Example..................................................................................................................115 Continue .............................................................................................................................116 Continue - Example ............................................................................................................117 Goto....................................................................................................................................118 Goto - Example...................................................................................................................118 Nested For Loops................................................................................................................120 Nested For Loops - Example...............................................................................................120

**C#.NET –Object Oriented Programming (OOP)...................................................................123** Introduction to Object Oriented Programming .................................................................123 Types of Programming Languages.....................................................................................123 Introduction of Object Oriented Programming (OOP).......................................................124 Object.................................................................................................................................124 Fields...................................................................................................................................125 Methods.............................................................................................................................125 Class....................................................................................................................................126 Reference variables............................................................................................................128 Syntax of creating class......................................................................................................128 Syntax of creating a Field in a class:...................................................................................128 Syntax of creating a method in a class:..............................................................................128 Syntax of creating reference variable ................................................................................129 Syntax of creating object (in heap): ...................................................................................129 Memory allocation of Objects............................................................................................129 Principles of Object Oriented Programming......................................................................130 Principles of OOP................................................................................................................130 Encapsulation:....................................................................................................................130 Abstraction:........................................................................................................................130

D. Harsha Vardhan (.NET Expert) P a g e 5 | 1583

**C#.NET 8.0**

Inheritance: ........................................................................................................................130 Polymorphism: ...................................................................................................................131 Types of polymorphism:.....................................................................................................131 Access Modifiers ................................................................................................................132 Access Modifiers ................................................................................................................132 Access Modifiers - Example................................................................................................135 Access Modifiers for classes...............................................................................................138 Access Modifiers for Classes ..............................................................................................138 Object Oriented Programming – Example .........................................................................138 Object Oriented Programming – Student Example ...........................................................140 Object Oriented Programming – Employee Example ........................................................142 Program.cs..........................................................................................................................143 Static Fields.........................................................................................................................145 Static Fields - Example........................................................................................................146 Constant Fields...................................................................................................................149 Constant Fields - Example ..................................................................................................149 ReadOnly Fields..................................................................................................................151 ReadOnly Fields - Example .................................................................................................151 Methods.............................................................................................................................153 Methods – Simple Example................................................................................................154 Methods – Arguments and Return Example......................................................................155 Scope of Variables..............................................................................................................157 Methods – Example - Age ..................................................................................................158 Methods – Example 2 - Numbers.......................................................................................160 Methods – Example 3 - Login.............................................................................................163 Methods – Example 4 - Student.........................................................................................165 "this" keyword....................................................................................................................170 "this" keyword - Example...................................................................................................170 Static Methods...................................................................................................................172 Static Methods - Example ..................................................................................................173 Reference Variables as Arguments....................................................................................175

D. Harsha Vardhan (.NET Expert) P a g e 6 | 1583

**C#.NET 8.0**

Reference Variables as Arguments - Example ...................................................................175 Reference Variables as Fields.............................................................................................177 Reference Variables as Fields - Example............................................................................177 Default Arguments.............................................................................................................179 Default Arguments - Example ............................................................................................179 Named Parameters ............................................................................................................181 Named Parameters - Example............................................................................................181 Methods Overloading.........................................................................................................183 Methods Overloading - Example........................................................................................183 Types of Parameters...........................................................................................................185 Call by value - Example.......................................................................................................186 Call by reference - Example................................................................................................187 Call by output - Example ....................................................................................................189 Type Conversion.................................................................................................................191 Implicit Casting - Example ..................................................................................................191 Explicit Casting (or) Type Casting .......................................................................................193 Explicit Casting (or) Type Casting - Example ......................................................................193 Parsing................................................................................................................................195 Parsing - Example ...............................................................................................................195 TryParse..............................................................................................................................197 TryParse - Example.............................................................................................................197 TryParse – Example 2 .........................................................................................................199 Output ................................................................................................................................200 Conversion Methods..........................................................................................................201 Conversion Methods - Example .........................................................................................202 Constructors.......................................................................................................................204 Types of Constructor..........................................................................................................204 Default Constructor............................................................................................................205 Parameterless Constructor - Example................................................................................205 Parameterized Constructor - Example...............................................................................207 Constructors - Student - Example ......................................................................................209

D. Harsha Vardhan (.NET Expert) P a g e 7 | 1583

**C#.NET 8.0**

Object Initializer.................................................................................................................212 Object Initializer - Example ................................................................................................212 "Set" and "Get" Methods...................................................................................................216 Set Method.........................................................................................................................216 Get Method........................................................................................................................216 "Set" and "Get" Methods - Example ..................................................................................217 Properties...........................................................................................................................219 Properties - Example ..........................................................................................................221 Readonly and WriteOnly Properties ..................................................................................223 Readonly Properties - Example ..........................................................................................224 Automatic Properties.........................................................................................................226 Automatic Properties - Example ........................................................................................226 Inheritance .........................................................................................................................228 Inheritance - Example ........................................................................................................228 Inheritance – Example 2.....................................................................................................231 “base” keyword..................................................................................................................234 “base” keyword - Example .................................................................................................234 Parent class’s constructor..................................................................................................237 Parent class’s constructor - Example .................................................................................237 Method Hiding ...................................................................................................................241 Method Hiding - Example...................................................................................................242 Method Hiding – Example 2...............................................................................................244 Method Overriding.............................................................................................................247 Normal Methods (vs) Virtual Methods..............................................................................247 Method Hiding (vs) Method Overrinding...........................................................................248 Method Overriding - Example............................................................................................249 Method Overriding – Example 2 ........................................................................................251 Abstract Classes..................................................................................................................254 Abstract Classes - Example.................................................................................................255 Abstract Classes – Example 2 .............................................................................................257 Abstract Methods...............................................................................................................260

D. Harsha Vardhan (.NET Expert) P a g e 8 | 1583

**C#.NET 8.0**

Abstract Methods - Example..............................................................................................262 Abstract Methods – Example 2 ..........................................................................................264 Interfaces............................................................................................................................268 Abstract Class (vs) Interface...............................................................................................268 Interfaces - Example...........................................................................................................271 Interfaces – Example 2 .......................................................................................................273 Dynamic Polymorphism .....................................................................................................276 Dynamic Polymorphism - Example.....................................................................................277 Dynamic Polymorphism – Example 2.................................................................................279 Multiple Inheritance using Interfaces................................................................................282 Multiple Inheritance using Interfaces - Example ...............................................................282 Interface Inheritance..........................................................................................................286 Interface Inheritance - Example.........................................................................................286 Sealed Classes.....................................................................................................................290 Sealed Classes - Example....................................................................................................290 Namespaces.......................................................................................................................293 Namespaces - Example ......................................................................................................293 Child Namespaces..............................................................................................................296 Child Namespaces - Example .............................................................................................296 Child Namespaces – Example 2..........................................................................................298 “using” statement ..............................................................................................................301 “using” statement - Example .............................................................................................301 “using - alias” statement....................................................................................................303 “using - alias” statement - Example ...................................................................................303 Creating Separate Files for Classes ....................................................................................306 Creating Separate Files for Classes - Example....................................................................306 Partial Classes.....................................................................................................................309 Partial Classes - Example....................................................................................................309 Enumerations.....................................................................................................................313 Enumerations - Example ....................................................................................................313 Structures...........................................................................................................................316

D. Harsha Vardhan (.NET Expert) P a g e 9 | 1583

**C#.NET 8.0**

Structures - Example ..........................................................................................................318 Structures with Constructors - Example ............................................................................320 Standard Data Types..........................................................................................................323 The "System.Object" class..................................................................................................324 “System.Object” Class - Example .......................................................................................324 Methods of "System.Object" class.....................................................................................326 Methods of “System.Object” class - Example....................................................................327 Boxing.................................................................................................................................330 Boxing - Example ................................................................................................................330 Unboxing ............................................................................................................................332 Unboxing - Example ...........................................................................................................332 Static Constructors.............................................................................................................334 Static Constructors - Example ............................................................................................335 Static Classes......................................................................................................................338 Static Classes - Example .....................................................................................................338 Generic Classes...................................................................................................................341 Generic Classes - Example..................................................................................................341 Multiple Generics - Example ..............................................................................................343 Arrays .................................................................................................................................346 Arrays - Example.................................................................................................................346 Arrays with For Loop ..........................................................................................................348 Arrays with For Loop - Example .........................................................................................348 Arrays with Foreach Loop in C#.NET ..................................................................................350 Arrays with Foreach Loop – Example.................................................................................350 “System.Array” class ..........................................................................................................353 “System.Array.IndexOf” method .......................................................................................354 “System.Array.IndexOf” method - Example.......................................................................354 “System.Array.IndexOf” method – with NotFound - Example ..........................................356 “System.Array.BinarySearch” method...............................................................................358 “System.Array.BinarySearch” method - Example ..............................................................358 “System.Array.Clear” method............................................................................................360

D. Harsha Vardhan (.NET Expert) P a g e 10 | 1583

**C#.NET 8.0**

“System.Array.Clear” method - Example...........................................................................360 “System.Array.Resize” method..........................................................................................363 “System.Array.Resize” method - Example .........................................................................363 “System.Array.Sort” method .............................................................................................366 “System.Array.Sort” method - Example.............................................................................366 “System.Array.Reverse” method .......................................................................................368 “System.Array.Reverse” method - Example ......................................................................368 “System.Array.CopyTo” method........................................................................................370 “System.Array.CopyTo” method - Example .......................................................................370 Multi-Dimensional Array ....................................................................................................373 Multi-Dimensional Array - Example ...................................................................................373 Collections..........................................................................................................................376 List of collections classes in .NET: ......................................................................................376 Arrays (vs) Collections........................................................................................................376 The "List" class....................................................................................................................377 The "List" class - Example...................................................................................................377 The "List" class – with “for” loop .......................................................................................380 The "List" class – with “for” loop - Example.......................................................................380 The "List" class – with “foreach” loop................................................................................382 The "List" class – with “foreach” loop - Example...............................................................382 "System.Collections.Generic. List.Add" method................................................................384 "System.Collections.Generic. List.Add" method - Example...............................................384 "System.Collections.Generic. List.Insert" method.............................................................386 "System.Collections.Generic.List.Insert" method - Example.............................................386 "System.Collections.Generic. List.AddRange" method......................................................388 "System.Collections.Generic. List.AddRange" method - Example.....................................388 "System.Collections.Generic. List.InsertRange" method...................................................391 "System.Collections.Generic. List.InsertRange" method - Example..................................391 "System.Collections.Generic. List.Remove" method in C#.NET.........................................394 "System.Collections.Generic. List.Remove" method - Example ........................................394 "System.Collections.Generic. List.RemoveAt" method .....................................................396

D. Harsha Vardhan (.NET Expert) P a g e 11 | 1583

**C#.NET 8.0**

"System.Collections.Generic. List.RemoveAt" method - Example.....................................396 "System.Collections.Generic. List.Clear" method..............................................................399 "System.Collections.Generic. List.Clear" method - Example .............................................399 "System.Collections.Generic. List.IndexOf" method .........................................................401 "System.Collections.Generic. List.IndexOf" method - Example.........................................401 "System.Collections.Generic. List.IndexOf" method – Not Found- Example.....................403 "System.Collections.Generic. List.BinarySearch" method .................................................405 "System.Collections.Generic. List.BinarySearch" method - Example ................................406 "System.Collections.Generic. List.Contains" method ........................................................407 "System.Collections.Generic. List.Contains" method - Example .......................................407 "System.Collections.Generic. List.Reverse" method .........................................................409 "System.Collections.Generic. List.Reverse" method - Example.........................................409 "System.Collections.Generic. List.Sort" method................................................................411 "System.Collections.Generic. List.Sort" method - Example...............................................411 System.Collections.Generic. List - Sort Descending - Example..........................................413 Collection Filter ..................................................................................................................415 Collection Filter - Example .................................................................................................415 LINQ....................................................................................................................................417 LINQ - Example...................................................................................................................417 Collection of Objects..........................................................................................................420 Collection of Objects - Example .........................................................................................420 Collection of Objects – Filter - Example .............................................................................422 Collection of Objects – LINQ - Example..............................................................................424 The "Dictionary" class ........................................................................................................427 The "Dictionary" class - Example........................................................................................427 The "SortedList" class.........................................................................................................429 The "SortedList" class - Example........................................................................................429 The "Hashtable" class in .NET ............................................................................................431 The "Hashtable" class - Example........................................................................................431 The "ArrayList" class in .NET ..............................................................................................433 The "ArrayList" class - Example..........................................................................................433

D. Harsha Vardhan (.NET Expert) P a g e 12 | 1583

**C#.NET 8.0**

The "typeof" operator in .NET............................................................................................435 The "typeof" operator - Example.......................................................................................435 The "System.String" class (or) String Handling ..................................................................439 “System.String.Length” - Example .....................................................................................442 “System.String.ToUpper” - Example ..................................................................................444 “System.String.ToLower” - Example ..................................................................................445 “System.String.GetChar” - Example...................................................................................447 “System.String.Substring” – Example 1 .............................................................................449 “System.String.Substring” – Example 2 .............................................................................450 “System.String.Remove” – Example ..................................................................................451 “System.String.Remove” – Example 2 ...............................................................................453 “System.String.Insert” – Example ......................................................................................455 “System.String.Equals” – Example .....................................................................................456 “System.StringComparison. OrdinalIgnoreCase” – Example .............................................458 “System.String.StartsWith” – Example ..............................................................................459 “System.String.EndsWith” – Example................................................................................461 “System.String.Contains” – Example .................................................................................462 “System.String.IndexOf” – Example 1................................................................................464 “System.String.IndexOf” – Example 2................................................................................465 “System.String.IndexOf” – Example 3................................................................................467 “System.String.LastIndexOf” – Example ............................................................................468 “System.String.Replace” – Example...................................................................................470 “System.String.ToCharArray” – Example ...........................................................................471 Converting CharArray to String - Example .........................................................................474 “System.String.Split” – Example.........................................................................................475 “System.String.Trim” – Example ........................................................................................477 “System.String.Format” – Example....................................................................................479 “System.String.Reverse” – Example...................................................................................480 String – WordsCount - Example .........................................................................................482 String – Character Occurrence Count - Example................................................................484 String – Alphabetic Count - Example..................................................................................486

D. Harsha Vardhan (.NET Expert) P a g e 13 | 1583

**C#.NET 8.0**

String – Word Occurrence Count - Example ......................................................................488 String – Title Case - Example ..............................................................................................490 String – Currency into words - Example.............................................................................492 String – Multiple Concatenations - Example......................................................................496 The "System.Text.StringBuilder" class...............................................................................498 The "System.Text.StringBuilder" class - Example ..............................................................499 The "System.DateTime" structure (or) Date & Time Handling ..........................................501 The "System.DateTime" – First Example...........................................................................503 The "System.DateTime.Now" –Example............................................................................505 The "System.DateTime" – Inner Values – Example...........................................................506 The "System.DateTime.DayOfWeek" – Example ...............................................................508 The "System.DateTime.DayOfYear" – Example .................................................................510 The "System.DateTime. ToShortDateString" – Example....................................................511 The "System.DateTime. ToLongDateString" – Example.....................................................513 The "System.DateTime. ToShortTimeString" – Example ...................................................514 The "System.DateTime. ToLongTimeString" – Example ....................................................516 The "System.DateTime" – Custom Formats – Example .....................................................517 The "System.DateTime.Subtract" – Example.....................................................................519 The "System.DateTime.AddDays" – Example ....................................................................521 The "System.Math" class....................................................................................................523 The "System.Math.Abs" –Example ....................................................................................524 The "System.Math.Floor" –Example ..................................................................................526 The "System.Math.Ceiling" – Example...............................................................................527 The "System.Math.Round" – Example 1............................................................................528 The "System.Math.Round" – Example 2............................................................................530 The "System.Math.Max" –Example ...................................................................................531 The "System.Math.Min" – Example ...................................................................................533 The "System.Math.Pow" –Example ...................................................................................534 The "System.Math.Sqrt" –Example....................................................................................536 “System.Diagnostics.Process” class ...................................................................................538 “System.Diagnostics.Process” class - Example ..................................................................538

D. Harsha Vardhan (.NET Expert) P a g e 14 | 1583

**C#.NET 8.0**

Command Line Arguments.................................................................................................540 Command Line Arguments - Example................................................................................540 Nullable Data Types............................................................................................................543 Nullable Data Types - Example...........................................................................................543 Exception Handling.............................................................................................................545 Exception Handling - Example............................................................................................546 Destructor ..........................................................................................................................549 Destructor - Example..........................................................................................................549 The "System.IDisposable" interface...................................................................................552 The "System.IDisposable" interface - Example..................................................................553 Garbage Collection.............................................................................................................555 Garbage Collection - Example ............................................................................................555 Delegates............................................................................................................................557 Single Cast Delegates - Example.........................................................................................558 Multi Cast Delegates - Example .........................................................................................560 Events.................................................................................................................................562 Events - Example ................................................................................................................563 Anonymous Methods.........................................................................................................566 Anonymous Methods - Example ........................................................................................566 Lambda Expressions...........................................................................................................569 Lambda Expressions - Example ..........................................................................................569 Inline Lambda Expressions.................................................................................................572 Inline Lambda Expressions - Example ................................................................................572 Inner Classes.......................................................................................................................575 Inner Classes - Example......................................................................................................575 Indexer................................................................................................................................578 Indexer - Example...............................................................................................................579 Assemblies..........................................................................................................................581 Class Libraries & Private Assemblies - Example .................................................................584 Shared Assemblies - Example.............................................................................................588 Documentation Comments................................................................................................593

D. Harsha Vardhan (.NET Expert) P a g e 15 | 1583

**C#.NET 8.0**

Documentation Comments - Example ...............................................................................593 Extension Methods ............................................................................................................596 Extension Methods - Example............................................................................................597

**C# – Console - System.IO Namespace..................................................................................601** Introduction to "System.IO" namespace ...........................................................................601 The “System.IO.FileInfo” class ...........................................................................................602 The “System.IO.FileInfo” class - Example...........................................................................605 The “System.IO.DirectoryInfo” class..................................................................................607 The “System.IO.DirectoryInfo” class - Example .................................................................610 The “System.IO.Directory” class ........................................................................................613 The “System.IO.Directory” class - Example........................................................................614 The “System.IO.File” class..................................................................................................617 The “System.IO.File” class - Example .................................................................................618 The “System.IO.FileStream” class......................................................................................621 The “System.IO.StreamWriter” class .................................................................................623 The “System.IO.StreamWriter” class - Example ................................................................624 The “System.IO.StreamReader” class ................................................................................626 The “System.IO.StreamReader” class - Example ...............................................................627

**C#.NET – Console –ADO.NET...................................................................................................629** Database Basics..................................................................................................................629 Introduction to ADO.NET ...................................................................................................631 List of pre-defined classes in ADO.NET ..............................................................................631 ADO.NET – “SqlConnection” class......................................................................................632 SqlConnection – Windows Authentication – Example.......................................................636 SqlConnection – SQL Server Authentication – Example ....................................................639 The "SqlCommand" class in ADO.NET................................................................................642 ADO.NET - ExecuteScalar ...................................................................................................646 SqlCommand – ExecuteScalar – Example ..........................................................................647 SqlCommand – ExecuteScalar – Example 2 .......................................................................650 ADO.NET – Connection Oriented Model – Introduction....................................................653

D. Harsha Vardhan (.NET Expert) P a g e 16 | 1583

**C#.NET 8.0**

The "SqlDataReader" class in ADO.NET .............................................................................654 ADO.NET – Connection Oriented Model............................................................................656 ADO.NET Connection Oriented Model – Single Record – Example ...................................657 ADO.NET Connection Oriented Model – Multiple Records – Example..............................661 The "SqlParameter" class in ADO.NET ...............................................................................665 ADO.NET Connection Oriented Model – SqlParameter – Example ...................................668 ADO.NET Disconnected Model...........................................................................................672 The “SqlDataAdapter” class ...............................................................................................673 The “DataSet” class............................................................................................................674 DataSet - Example ..............................................................................................................674 ADO.NET Disconnected Model – Example.........................................................................679 ADO.NET Disconnected Model – Multiple Tables - Example.............................................683 ADO.NET Disconnected Model – Joins - Example..............................................................688 ADO.NET Non Query - Insertion – Example .......................................................................692 ADO.NET Non-Query - Updation – Example ......................................................................695 ADO.NET Non-Query - Deletion – Example........................................................................698 ADO.NET Non Query – Insertion – With SqlParameter – Example....................................701 ADO.NET Non-Query – Updation – With SqlParameter – Example...................................705 ADO.NET Non-Query – Deletion – With SqlParameter – Example ....................................709 Stored Procedures Calling in ADO.NET ..............................................................................712 ADO.NET Non Query – Insertion – With Stored Procedures – Example............................713 ADO.NET Non Query – Updation – With Stored Procedures – Example...........................717 ADO.NET Non Query – Deletion – With Stored Procedures – Example ............................721 ADO.NET - Transactions.....................................................................................................725 ADO.NET – Transactions - Example....................................................................................726 The "OleDb" namespace in ADO.NET ................................................................................730 Connection Strings in ADO.NET..........................................................................................731 ADO.NET – Oracle - Example..............................................................................................732 ADO.NET – MS Access - Example .......................................................................................743 ADO.NET – MS Excel - Example..........................................................................................755 ADO.NET – SQL Server to Oracle - Example.......................................................................764

D. Harsha Vardhan (.NET Expert) P a g e 17 | 1583

**C#.NET 8.0**

ADO.NET – SQL Server to MS Excel - Example...................................................................771 ADO.NET – Oracle to SQL Server - Example.......................................................................779 ADO.NET – Excel to SQL Server - Example .........................................................................786 ADO.NET – SQL Server to File - Example............................................................................794 ADO.NET – File to SQL Server - Example............................................................................799 The "SqlCommandBuilder" class in ADO.NET ....................................................................804 ADO.NET – SqlCommandBuilder – DataSet – Insertion - Example ....................................805 ADO.NET – SqlCommandBuilder – DataSet – Updation - Example....................................809 ADO.NET – SqlCommandBuilder – DataSet – Deletion - Example.....................................813 N-Tier Architecture.............................................................................................................817 N-Tier Architecture - Example............................................................................................817 LINQ to SQL ........................................................................................................................825 LINQ to SQL - Example........................................................................................................826

**C#.NET – Console – Entity Framework..................................................................................831** Intro to ADO.NET Entity Framework..................................................................................831 Entity Framework - Example ..............................................................................................834 Entity Framework – FirstOrDefault - Example ...................................................................839 Entity Framework – Insertion - Example............................................................................844 Entity Framework – Updation - Example ...........................................................................849 Entity Framework – Deletion - Example ............................................................................854

**C#.NET – Windows Forms Applications................................................................................858** Introduction to Windows Forms Applications...................................................................858 The “System.Windows.Forms.Form” class ........................................................................859 Programming Model of Windows Form.............................................................................860 Windows Forms Application – First Example.....................................................................861 Form Constructor - Example ..............................................................................................863 Properties of “System.Windows.Forms.Form” class .........................................................865 “System.Windows.Forms.Form. Text” property - Example...............................................867 “System.Windows.Forms.Form. ShowIcon” property - Example ......................................869 “System.Windows.Forms.Form. ShowInTaskBar” property -Example ..............................871

D. Harsha Vardhan (.NET Expert) P a g e 18 | 1583

**C#.NET 8.0**

“System.Windows.Forms.Form. MinimizeBox” property - Example.................................872 “System.Windows.Forms.Form. MaximizeBox” property - Example ................................874 “System.Windows.Forms.Form. ControlBox” property - Example....................................876 “System.Windows.Forms.Form. TopMost” property - Example .......................................877 “System.Windows.Forms.Form. WindowState” property –Example................................879 “System.Windows.Forms.Form. FormBorderStyle” property – Example..........................880 “System.Windows.Forms.Form. Cursor” property - Example ...........................................882 “System.Windows.Forms.Form. BackColor” property - Example......................................884 “System.Windows.Forms.Form. BackgroundImage” property – Example........................885 “System.Windows.Forms.Form. Size” property - Example................................................887 “System.Windows.Forms.Form. Location” property - Example ........................................889 “System.Windows.Forms.Form. Icon” property - Example...............................................890 Events of “System.Windows.Forms.Form” class ...............................................................892 “System.Windows.Forms.Form. Load” event - Example ...................................................893 “System.Windows.Forms.Form. Shown” event - Example................................................896 “System.Windows.Forms.Form. Click” event - Example ...................................................897 “System.Windows.Forms.Form. DoubleClick” event - Example........................................899 “System.Windows.Forms.Form.MouseClick” event - Example .........................................901 “System.Windows.Forms.Form. MouseMove” event - Example ......................................903 “System.Windows.Forms.Form. KeyPress” event - Example.............................................904 “System.Windows.Forms.Form. FormClosing” event - Example.......................................906 “System.Windows.Forms.Form. FormClosing” event - Example 2....................................908 Methods of “System.Windows.Forms.Form” class............................................................910 “System.Windows.Forms.Form. Hide” method - Example................................................911 “System.Windows.Forms.Form .Show” method - Example ..............................................913 “System.Windows.Forms.Form. Close” method - Example...............................................915 Introduction to Windows Forms Controls .........................................................................917 “System.Windows.Forms.Label” class...............................................................................919 “System.Windows.Forms.Label” class - Example ..............................................................922 “System.Windows.Forms. Button” class............................................................................925 “System.Windows.Forms. Button” class - Example...........................................................929

D. Harsha Vardhan (.NET Expert) P a g e 19 | 1583

**C#.NET 8.0**

“System.Windows.Forms. Button” class – with Image - Example.....................................932 “System.Windows.Forms. Button” class – with Label - Example ......................................934 “System.Windows.Forms. TextBox” class..........................................................................938 “System.Windows.Forms. TextBox” class - Example .........................................................943 “System.Windows.Forms. TextBox” class – With AutoComplete - Example.....................947 “System.Windows.Forms. TextBox” class – With AutoComplete – Example 2 .................949 “System.Windows.Forms. TextBox” class – With KeyPress – Example .............................952 “System.Windows.Forms. TextBox” class – With KeyPress – Example 2 ..........................955 “System.Windows.Forms. TextBox” class – With Enter and Leave – Example..................957 “System.Windows.Forms. TextBox” class – With TabIndex – Example.............................961 “System.Windows.Forms. TextBox” class – Math – Example............................................965 “System.Windows.Forms. NumericUpDown” class...........................................................970 “System.Windows.Forms. NumericUpDown” class – Example .........................................973 “System.Windows.Forms. DateTimePicker” class .............................................................976 “System.Windows.Forms. DateTimePicker” class – Example............................................980 “System.Windows.Forms. MonthCalendar” class .............................................................983 “System.Windows.Forms. MonthCalendar” class – Example............................................987 “System.Windows.Forms. ToolTip” class...........................................................................991 “System.Windows.Forms. ToolTip” class – Example .........................................................993 “System.Windows.Forms. ErrorProvider” class.................................................................995 “System.Windows.Forms. ErrorProvider” class – Example ...............................................997 “System.Text. RegularExpressions. Regex” class.............................................................1000 “System.Text. RegularExpressions. Regex” class – Example............................................1001 “System.Windows.Forms. MaskedTextBox” class...........................................................1005 “System.Windows.Forms. MaskedTextBox” class - Example ..........................................1008 “System.Windows.Forms. CheckBox” class.....................................................................1011 “System.Windows.Forms. CheckBox” class - Example ....................................................1014 “System.Windows.Forms. RadioButton” class.................................................................1016 “System.Windows.Forms. RadioButton” class - Example................................................1019 “System.Windows.Forms. ComboBox” class...................................................................1023 “System.Windows.Forms. ComboBox” class - Example ..................................................1026

D. Harsha Vardhan (.NET Expert) P a g e 20 | 1583

**C#.NET 8.0**

“Cascading ComboBox” - Example...................................................................................1029 “System.Windows.Forms. ListBox” class .........................................................................1032 “System.Windows.Forms. ListBox” class - Example.........................................................1035 “System.Windows.Forms. CheckedListBox” class............................................................1038 “System.Windows.Forms. CheckedListBox” class – Example..........................................1041 “System.Windows.Forms. TreeView” class......................................................................1044 “System.Windows.Forms. TreeView” class - Example.....................................................1049 The “System.Windows.Forms. PictureBox” class ............................................................1052 “System.Windows.Forms. PIctureBox” class - Example ..................................................1055 “System.Windows.Forms. PIctureBox” class – Example 2...............................................1057 The “System.Windows.Forms. Panel” class.....................................................................1061 “System.Windows.Forms. Panel” class - Example ...........................................................1063 The “System.Windows.Forms. GroupBox” class..............................................................1066 “System.Windows.Forms. GroupBox” class - Example....................................................1069 The “System.Windows.Forms. SplitContainer” class.......................................................1072 “System.Windows.Forms. SplitContainer” class - Example.............................................1075 The “System.Windows.Forms. TabControl” class............................................................1077 “System.Windows.Forms. TabControl” class - Example ..................................................1080 The “System.Windows.Forms. FlowLayoutPanel” class ..................................................1083 “System.Windows.Forms. FlowLayoutPanel” class - Example ........................................1086 “System.Windows.Forms. LinkLabel” class......................................................................1088 “System.Windows.Forms. LinkLabel” class - Example.....................................................1091 “System.Windows.Forms. WebBrowser” class................................................................1094 “System.Windows.Forms. WebBrowser” class - Example...............................................1096 “System.Windows.Forms.Timer” class ............................................................................1099 “System.Windows.Forms. Timer” class - Example...........................................................1101 “System.Windows.Forms.Timer” class – with Time - Example.......................................1104 “System.Windows.Forms.Timer” class – with Counter - Example..................................1106 “System.Windows.Forms.Timer” class – with Slide Show - Example .............................1108 “System.Windows.Forms. ProgressBar” class .................................................................1112 “System.Windows.Forms. ProgressBar” class - Example.................................................1115

D. Harsha Vardhan (.NET Expert) P a g e 21 | 1583

**C#.NET 8.0**

“System.Windows.Forms. NotifyIcon” class....................................................................1118 “System.Windows.Forms. NotifyIcon” class - Example ...................................................1120 Popup Boxes.....................................................................................................................1123 PopupBoxes - Example.....................................................................................................1123 “System.Windows.Forms. ColorDialog” class..................................................................1130 “System.Windows.Forms. ColorDialog” class - Example .................................................1131 “System.Windows.Forms. FontDialog” class ...................................................................1135 “System.Windows.Forms. FontDialog” class - Example...................................................1137 “System.Windows.Forms. FolderBrowserDialog” class...................................................1140 “System.Windows.Forms. FolderBrowserDialog” class - Example..................................1143 “System.Windows.Forms. OpenFileDialog” class............................................................1146 “System.Windows.Forms. OpenFileDialog” class - Example ...........................................1148 “System.Windows.Forms. SaveFileDialog” class..............................................................1152 “System.Windows.Forms. SaveFileDialog” class - Example.............................................1154 “System.Windows.Forms. MenuStrip” class....................................................................1158 “System.Windows.Forms. MenuStrip” class - Example...................................................1163 “System.Windows.Forms. ContextMenuStrip” class.......................................................1167 “System.Windows.Forms. ContextMenuStrip” class - Example ......................................1170 “System.Windows.Forms. ToolStrip” class......................................................................1174 “System.Windows.Forms. ToolStrip” class - Example .....................................................1178 “System.Windows.Forms. StatusStrip” class...................................................................1183 “System.Windows.Forms. StatusStrip” class - Example ..................................................1187 “System.Windows.Forms. RichTextBox” class.................................................................1190 “System.Windows.Forms. RichTextBox” class - Example ................................................1195 User Controls....................................................................................................................1208 User Controls - Example...................................................................................................1209 Windows Forms Control Library.......................................................................................1214 Windows Forms Control Library - Example......................................................................1214 “System.Net.Mail.SmtpClient” class................................................................................1220 “System.Net.Mail.SmtpClient” class - Example ...............................................................1221 Multi-Threading................................................................................................................1225

D. Harsha Vardhan (.NET Expert) P a g e 22 | 1583

**C#.NET 8.0**

Multi Threading - Example ...............................................................................................1229 Task Parallel Library..........................................................................................................1231 Task Parallel Library - Example.........................................................................................1233 Windows Services.............................................................................................................1236 Windows Services - Example............................................................................................1237

**C#.NET – WinForms – System.IO Namespace...................................................................1242** The “System.IO.FileInfo” class - Example.........................................................................1242 The “System.IO.DirectoryInfo” class - Example ...............................................................1246 The “System.IO.Directory” class - Example......................................................................1252 The “System.IO.File” class - Example ...............................................................................1255 The “System.IO.StreamWriter” class - Example ..............................................................1260 The “System.IO.StreamReader” class - Example .............................................................1263

**C#.NET – WinForms – ADO.NET..............................................................................................1267** SqlConnection – Windows Authentication – Example.....................................................1267 SqlConnection – SQL Server Authentication – Example ..................................................1270 SqlCommand – ExecuteScalar – Example ........................................................................1274 SqlCommand – ExecuteScalar – Example 2 .....................................................................1277 ADO.NET Connection Oriented Model – Single Record – Example .................................1281 ADO.NET Connection Oriented Model – Multiple Records – Example............................1285 ADO.NET Connection Oriented Model – Multiple Records - Label – Example................1290 ADO.NET Connection Oriented Model – SqlParameter – Example .................................1294 ADO.NET Connection Oriented Model – SqlParameter – ComboBox – Example............1299 DataSet - Example ............................................................................................................1305 ADO.NET Disconnected Model – Example.......................................................................1311 ADO.NET Disconnected Model – Multiple Tables - Example...........................................1316 ADO.NET Disconnected Model – Joins - Example............................................................1322 ADO.NET Disconnected Model – Record Navigations - Example ....................................1327 ADO.NET Disconnected Model – ComboBox - Example ..................................................1334 ADO.NET Disconnected Model – ComboBox - DataSource - Example ............................1340 ADO.NET Disconnected Model – DataGridView - Example .............................................1347

D. Harsha Vardhan (.NET Expert) P a g e 23 | 1583

**C#.NET 8.0**

ADO.NET Disconnected Model – DataGridView - SqlCommandBuilder - Example.........1350 ADO.NET Disconnected Model – Master – Child – Example ...........................................1354 ADO.NET Disconnected Model – Cascading ComboBox – Example ................................1360 ADO.NET Non Query - Insertion – Example .....................................................................1366 ADO.NET Non Query - Updation – Example.....................................................................1369 ADO.NET Non Query - Deletion – Example......................................................................1372 ADO.NET Non Query – Insertion – With SqlParameter – Example..................................1376 ADO.NET Non Query – Updation – With SqlParameter – Example .................................1381 ADO.NET Non Query – Deletion – With SqlParameter – Example ..................................1386 ADO.NET Non Query – Insertion – With Stored Procedures – Example..........................1390 ADO.NET Non Query – Updation – With Stored Procedures – Example.........................1395 ADO.NET Non Query – Deletion – With Stored Procedures – Example ..........................1401 ADO.NET Disconnected Model – ComboBox – Stored Procedures – Example................1405 ADO.NET Disconnected Model – Updation with ComboBox - Example..........................1411 ADO.NET – CRUD (Create, Retrieve, Update, Delete) - Example.....................................1418 ADO.NET – Registration Form - Example .........................................................................1432 ADO.NET – Login Form - Example ....................................................................................1439 ADO.NET – Transactions - Example..................................................................................1445 ADO.NET – Oracle - Example............................................................................................1449 ADO.NET – MS Access - Example .....................................................................................1461 ADO.NET – MS Excel - Example........................................................................................1475 ADO.NET – SQL Server to Oracle - Example.....................................................................1484 ADO.NET – SQL Server to MS Excel - Example.................................................................1491 ADO.NET – Oracle to SQL Server - Example.....................................................................1500 ADO.NET – Excel to SQL Server - Example .......................................................................1508 ADO.NET – SQL Server to File - Example..........................................................................1516 ADO.NET – File to SQL Server - Example..........................................................................1522 ADO.NET – SqlCommandBuilder – DataSet – Insertion - Example ..................................1528 ADO.NET – SqlCommandBuilder – DataSet – Updation - Example..................................1533 ADO.NET – SqlCommandBuilder – DataSet – Deletion - Example...................................1538 N-Tier Architecture - Example..........................................................................................1542

D. Harsha Vardhan (.NET Expert) P a g e 24 | 1583

**C#.NET 8.0**

LINQ to SQL - Example......................................................................................................1551 **C#.NET – WinForms – Entity Framework...........................................................................1559** Entity Framework - Example ............................................................................................1559 Entity Framework – Insertion - Example..........................................................................1565 Entity Framework – Updation - Example .........................................................................1571 Entity Framework – Deletion - Example ..........................................................................1577

D. Harsha Vardhan (.NET Expert) P a g e 25 | 1583

**C#.NET 8.0**

1

**.NET Fundamentals**

Introduction to .NET

What is .NET?

• .NET is an “application development platform”, which is used to develop desktop, web and mobile applications.

• “Application” is a program (collection of instructions) that can run based on the operating system.

• “Application Development” is a process of creating new applications based on the client’s requirements.

• “Application Development Platform” is a software tool, based on which you can develop applications.

• .NET is developed by Microsoft Corporation in 2002.

• “NET” is named to mean “Internet Age” applications; it has not abbreviation.

Parts of .NET

• .NET is mainly divided into 3 parts:

D. Harsha Vardhan (.NET Expert) P a g e 26 | 1583

**C#.NET 8.0**

1. C#.NET (C Sharp .NET)

2. ASP.NET (Active Server Pages .NET)

3. Xamarin.NET

1. C#.NET

• Covers language basics.

• It is used to develop “Stand-alone applications” that runs based on the single machine.

• Ex: Calculator, Bill Generation Applications etc.

2. ASP.NET

• Covers server side programming.

• It is used to develop “Web applications” that runs based on “Client-server architecture”.

3. Xamarin.NET

• Covers mobile programming.

• It is used to develop “Mobile applications” that runs based on “Mobile devices”.

History of .NET

History of .NET

• 1997: Microsoft wants to create a better programming language to develop standalone

and client-server applications. Microsoft team (led by Anders Hejlsberg, founder of Turbo Pascal) has started developing the new language.

D. Harsha Vardhan (.NET Expert) P a g e 27 | 1583

**C#.NET 8.0**

• 1998: Microsoft has completed developing the new language and named it as Simple Managed C (SMC).

• 1999: Microsoft renamed “SMS” as “C#” and wants to integrate C#, ASP. They named the integrated development platform as “NGWS” (Next Generation Windows Services).

• 2000: Microsoft renamed “NGWS” as “.NET” (Network-based). Microsoft announced “.NET” in July 2000.

• 2001: Microsoft has integrated C#.NET and ASP.NET in “.NET”.

• 2002: .NET 1.0

• 2003: .NET 1.1

• 2005: .NET 2.0

• 2006: .NET 3.0

• 2007: .NET 3.5

• 2010: .NET 4.0

• 2012: .NET 4.5

• 2013: .NET 4.5.1

• 2014: .NET 4.5.2

• 2015: .NET 4.6

• 2015: .NET 4.6.1

• 2016: .NET 4.6.2

• 2017: .NET 4.7

• 2017: .NET 4.7.1

• 2018: .NET 4.7.2

• 2019: .NET 4.8

D. Harsha Vardhan (.NET Expert) P a g e 28 | 1583

**C#.NET 8.0**

Versions of .NET

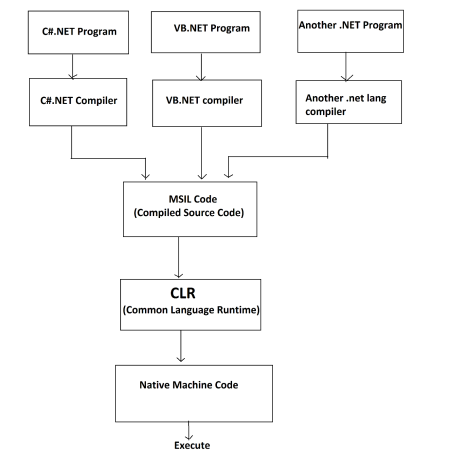
| Sl.  No | .NET Framework | Year of  release | Concepts |
| --- | --- | --- | --- |
| 1 | .NET Framework 1.0 | 2002 | Console Apps,  Language Fundamentals,  OOP,  WinForms,  Web Forms,  ADO.NET |
| 2 | .NET Framework 1.1 | 2003 | ASP.NET Security |
| 3 | .NET Framework 2.0 | 2005 | Improved web controls,  Data controls, Themes and skins,  Master pages,  Partial classes, Nullable types,  Anonymous methods, Generics |
| 4 | .NET Framework 3.0 | 2006 | WPF  WCF |
| 5 | .NET Framework 3.5 | 2007 | LINQ  ASP.NET AJAX |
| 6 | .NET Framework 4.0 | 2010 | ADO.NET Entity Framework  Task Parallel Library  ASP.NET Web Pages  ASP.NET MVC |
| 7 | .NET Framework 4.5 | 2012 | Open Authentication  ASP.NET Web API  ASP.NET Bundles and Minification |
| 8 | .NET Framework 4.5.1 | 2013 | Async and Await |
| 9 | .NET Framework 4.5.2 | 2014 | Bug fixes |
| 10 | .NET Framework 4.6 | 2015 | .NET Core  ASP.NET Core |
| 11 | .NET Framework 4.6.1 | 2015 | Spell Check for WPF |
| 12 | .NET Framework 4.6.2 | 2016 | Soft keyboard support for WPF |
| 13 | .NET Framework 4.7 | 2017 | Print API for WPF |
| 14 | .NET Framework 4.8 | 2019 | Performance and Security Updates |

D. Harsha Vardhan (.NET Expert) P a g e 29 | 1583

**C#.NET 8.0**

CIL and CLR

Execution Model of .NET Programs

Steps:

1. Source Code: The source code of the program is written in “.cs” file.

2. Compilation Process: “CSC” (C Sharp Compiler) compiles (converts) the program from “C#.NET” (programmer understandable representation) to “MSIL (Microsoft Intermediate Language)” (also called as “byte code”). The “MSIL code” will be saved in “.exe” file. This

D. Harsha Vardhan (.NET Expert) P a g e 30 | 1583

**C#.NET 8.0**

EXE file can’t execute directly. The MSIL is neither understandable by the programmer, nor by operating system.

3. Execution Process: CLR (Common Language Runtime) converts the program from MSIL language at “EXE file” to “native machine language” (based on the current operating system). Operating System executes the “native machine language”. Then we will get output.

CIL (Common Intermediate Language) or MSIL

▪ CIL or MSIL (Microsoft Intermediate Language) is the intermediate language, developed by Microsoft Corporation, for .NET.

▪ The .net programs are converted into “MSIL language” first; and then converted into “native machine language”.

▪ The MSIL code will be stored in "EXE" file. Ex: filename.exe. The “native machine language” code will not be saved in any file, it directly runs.

▪ Once the code is converted into MSIL, it doesn't matter in which .net language it is originally developed. So for all .net languages, we can have a common runtime engine called "CLR".

CLR (Common Language Runtime)

▪ CLR stands for “Common Language Runtime”.

▪ CLR is the “Execution Engine” or “Execution Environment” of .NET.

▪ To run any type of .net program (app), CLR must be installed in the computer.

▪ CLR will be started automatically when the .net application execution starts. CLR performs essential tasks internally, while running any .net application.

▪ Without CLR, we can’t run any .net program.

▪ CLR reads the “MSIL code” from the EXE file, converts the same into “native machine code”, gives the same to the operating system and then operating system executes the native

D. Harsha Vardhan (.NET Expert) P a g e 31 | 1583

**C#.NET 8.0**

machine code; then the user gets the output. So CLR is helps the operating system while executing the .net program.

▪ CLR will be installed automatically as a part of “.NET Framework” software. CLR must be installed in both developer’s machine and user’s machine.

Sub Components of CLR

▪ CLR has the following inner components (sub components).

1. Memory Manager: “Memory Manager” is a sub component in CLR, which allocates memory (in RAM) for the variables and objects in the program.

2. Garbage Collector: “Garbage Collector” is a sub component in CLR, which deletes the variables and objects that are created during the program, automatically at the end of the program execution.

3. Class Loader: “Class Loader” is a sub component in CLR, which loads a class on-demand. When we try to access a class in the program for the first time, then the “Class Loader” searches for the class in the entire program, loads the class into the memory, and it passes the class to JIT compiler. That means if we don’t call a class, it will not be loaded into memory. This avoids un-necessary loading of the classes, if we don’t require them. So it improves performance.

4. JIT (Just-In-Time) Compiler: “JIT Compiler” is a sub component in CLR, which converts the “MSIL code” into “native machine language” (based on current operating system).

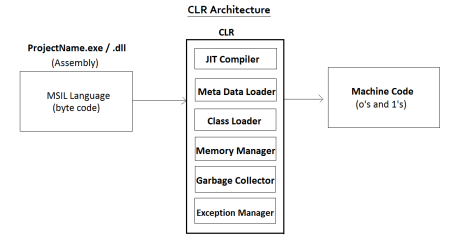
5. Thread Manager: “Thread Manager” is a sub component in CLR, which manages the threads of the program. It gives necessary instructions to the processor, which thread is to be executed when. A thread is a "part of the program" or "background work".

D. Harsha Vardhan (.NET Expert) P a g e 32 | 1583

**C#.NET 8.0**

6. Exception Manager: “Exception Manager” is a sub component in CLR, which passes necessary instructions to the operating system, which code should be executed when an exception (runtime error) occurs while executing the program.

7. Security Manager: “Security Manager” is a sub component in CLR, which takes care about different types of security in .net such as windows authentication, forms authentication, open authentication etc.



D. Harsha Vardhan (.NET Expert) P a g e 33 | 1583

**C#.NET 8.0**

.NET Framework Architecture

What is .NET Framework

• It is the “Software Development Kit (SDK)”, which contains many components such as CIL, CLR, FCL, CLS, which is used to develop & run the .net applications.

• It should be installed in both developer system and user system.

• .NET Framework download link:

https://www.microsoft.com/en-us/download/details.aspx?id=55167

System Requirements of .NET Framework:

| Software / Hardware Requirement | Minimum | Recommended |
| --- | --- | --- |
| Processor | 1.6 Ghz | 2.3 Ghz or higher |
| RAM | 1 GB | 2 GB or higher |
| Hard disk free space on C: drive | 2 GB | 4 GB or higher |
| Operating System | Windows 7 + Service Pack 1 Windows 8  Windows 8.1  Windows 10 | Windows 10 |

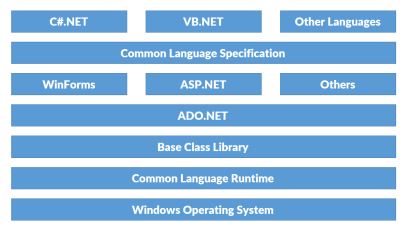
D. Harsha Vardhan (.NET Expert) P a g e 34 | 1583

**C#.NET 8.0**

.NET Framework Architecture

• “.NET framework” is divided as several components.

• “.NET Framework Architecture” explains the list of components of .NET Framework, and how they are arranged internally.

**Explanation:**

• "Windows" operating system works based on "processor".

• "CLR" works based on "Windows Operating System".

• "BCL" works based on "CLR".

• "ADO.NET" works based on "BCL".

• "WinForms", "ASP.NET", and “Other Frameworks” work based on "ADO.NET" and "BCL". • "CLS" is created based on "WinForms", "ASP.NET" and “Other Frameworks”. • "C#.NET", "VB.NET", "Other Languages" are developed based on "CLS".

• Finally, the programmers are writing the programs by using the languages called “C#.NET, “VB.NET” etc.

D. Harsha Vardhan (.NET Expert) P a g e 35 | 1583

**C#.NET 8.0**

CLS and CTS

CLS (Common Language Specification)

▪ “CLS” is the set of rules, based on which all .net languages (C#.NET, VB.NET, VC++.NET etc.) are developed.

▪ The common rules are about literals, operators, identifiers, data types, type conversion, object oriented programming etc.

CTS (Common Type System)

▪ “CTS” is a set common data types, based on which, the data types of all .net languages (C#.NET, VB.NET, VC++.NET etc.) are developed.

▪ Thus, we achieve the uniform data types among all .net languages.

▪ The following is the list of data types of CTS:

1. SByte

2. Byte

3. Short

4. UShort

5. Int32

6. UInt32

7. Int64

8. UInt64

9. Single

10. Double

11. Decimal

12. Char

13. String

14. Boolean

D. Harsha Vardhan (.NET Expert) P a g e 36 | 1583

**C#.NET 8.0**

FCL and BCL

FCL (Framework Class Library)

▪ .NET provides a set of classes and interfaces, based on which we can develop .net applications.

▪ The “DLL (Dynamic Link Library)” file is a collection of namespaces; Namespace is a collection of classes and interfaces.

▪ FCL is divided into the following parts:

1. BCL: BCL is a set of classes and interfaces, which can be used in all types of applications.

2. WinForms: This is a set of classes and interfaces, which can be used only in windows applications.

3. ASP.NET: This is a set of classes and interfaces, which can be used only in web applications.

4. ADO.NET: This is a set of classes and interfaces, which can be used in all types of applications for connecting to databases.

D. Harsha Vardhan (.NET Expert) P a g e 37 | 1583

**C#.NET 8.0**

Introduction to Visual Studio

What is Visual Studio

• Visual Studio is the “IDE” (Integrated Development Environment), where you can write all types of .net programs (C#.NET, ASP.NET).

• It provides advantages such as syntax highlighting, intellisense, deployment etc.

Versions of Visual Studio

| Visual Studio Version | Year of  Release | Supported Operating Systems | Supported  .NET  Framework  Versions |
| --- | --- | --- | --- |
| Visual Studio 2002 (7.0) | 2002 | Windows 2000  Windows XP | .NET 1.0 |
| Visual Studio 2003 (7.1) | 2003 | Windows 2000  Windows XP | .NET 1.1 |
| Visual Studio 2005 (8.0) | 2005 | Windows 2000  Windows XP | .NET 2.0 |
| Visual Studio 2008 (9.0) | 2007 | Windows 2000  Windows XP  Windows 7 | .NET 2.0  .NET 3.0  .NET 3.5 |
| Visual Studio 2010 (10.0) | 2010 | Windows XP + Service Pack 3  Windows 7 | .NET 2.0  .NET 3.0  .NET 3.5  .NET 4.0 |
| Visual Studio 2012 (11.0) | 2012 | Windows 7  Windows 8.1  Windows 10 | .NET 2.0  .NET 3.0  .NET 3.5  .NET 4.0  .NET 4.5 |
| Visual Studio 2013 (12.0) | 2013 | Windows 7 + Service Pack 1  Windows 8.1 | .NET 2.0  .NET 3.0 |

D. Harsha Vardhan (.NET Expert) P a g e 38 | 1583

**C#.NET 8.0**

|  |  | Windows 10 | .NET 3.5  .NET 4.0  .NET 4.5  .NET 4.5.1 |
| --- | --- | --- | --- |
| Visual Studio 2015 (14.0) | 2015 | Windows 7 + Service Pack 1  Windows 8.1  Windows 10 | .NET 2.0  .NET 3.0  .NET 3.5  .NET 4.0  .NET 4.5  .NET 4.5.1  .NET 4.5.2  .NET 4.6  .NET 4.6.1  .NET 4.6.2 |
| Visual Studio 2017 (15.0) | 2017 | Windows 7 + Service Pack 1  Windows 8.1  Windows 10 | .NET 2.0  .NET 3.0  .NET 3.5  .NET 4.0  .NET 4.5  .NET 4.5.1  .NET 4.5.2  .NET 4.6  .NET 4.6.1  .NET 4.6.2  .NET 4.7 |
| Visual Studio 2019 |  |  |  |

System Requirements of Visual Studio 2019:

| Software / Hardware Requirement | Minimum | Recommended |
| --- | --- | --- |
| Processor | 1.6 Ghz | 2.3 Ghz or higher |
| RAM | 2 GB | 4 GB or higher |
| Hard disk free space on | 10 GB | 15 GB or higher |

D. Harsha Vardhan (.NET Expert) P a g e 39 | 1583

**C#.NET 8.0**

| C: drive |  |  |
| --- | --- | --- |
| Operating System | Windows 7 + Service Pack 1 Windows 8.1  Windows 10 | Windows 10 |
| Internet Explorer | Internet Explorer 10 | Internet Explorer 11 or above |

Project

• Project is a folder, which is a “group of files”.

• A file contains code (program).

• When we compile the project, Visual Studio generates only one EXE file for the entire project, which contains the compiled source code (in MSIL language) of all the files of the same project.

Solution

• Solution is a folder, which is a “group of projects”.

File Types of .NET

• .NET supports the following file types:

| File Extension | Full form | Description |
| --- | --- | --- |
| .cs | C# file | A C# file contains the C#.NET program (source code). |
| .vb | VB file | A VB file contains the VB.NET program (source code). |
| .csproj | C# Project | A project is a collection of files.  The project file contains the list of all the files in the current project. |
| .sln | Solution | A solution is a collection of projects.  The solution file contains the list of all the projects in the current solution. |

D. Harsha Vardhan (.NET Expert) P a g e 40 | 1583

**C#.NET 8.0**

| .exe | Executable file | The EXE file contains the compiled source code of a project. For every project, a separate EXE file will be created. |
| --- | --- | --- |
| .config | Configuration File | The configuration file contains the configuration settings of a project. |

Folder Structure of .NET Project in Visual Studio • .NET program (project) should have the following folder structure. • Every solution is a folder. Every project is folder.

**Solution Folder**

✧ **Project Folder**

o filename.cs

o App.config

o projectname.csproj

o bin\projectname.exe

✧ **solutionname.sln**

D. Harsha Vardhan (.NET Expert) P a g e 41 | 1583

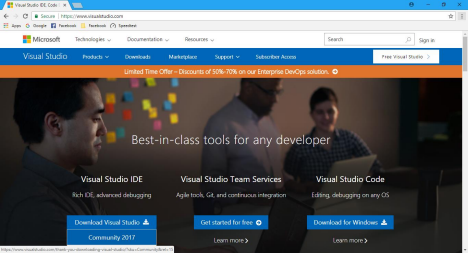
**C#.NET 8.0**

Visual Studio - Installation

Steps to install “Visual Studio 2019”:

Go to http://www.visualstudio.com

Click on “Download Visual Studio” – “Community 2019”.



You will get “vs\_community.exe” file.

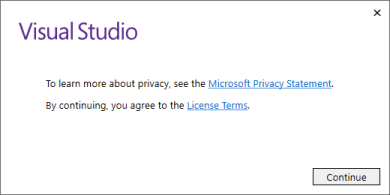
Run “vs\_community.exe” file.

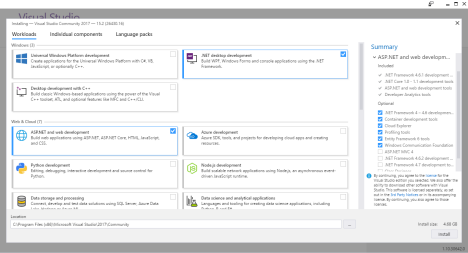
Click on “Run”.

Click on “Continue”.

D. Harsha Vardhan (.NET Expert) P a g e 42 | 1583

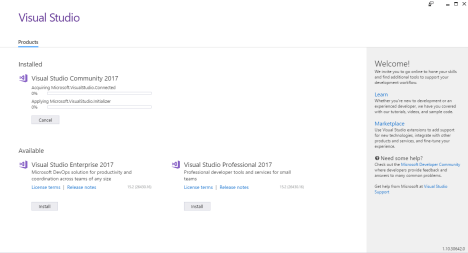
**C#.NET 8.0**

****

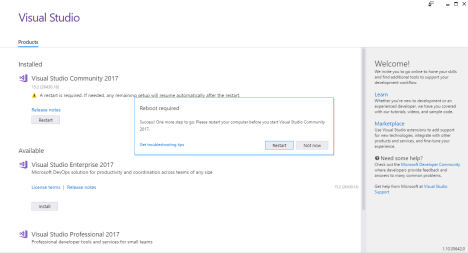
Check the checkboxes “.NET desktop development” and “ASP.NET and web development”. Click on “Install”.

D. Harsha Vardhan (.NET Expert) P a g e 43 | 1583

**C#.NET 8.0**

****

Wait for installation to be completed.

Click on “Restart”.

After restarting your system, go to “Start” > “Visual Studio 2019”.

D. Harsha Vardhan (.NET Expert) P a g e 44 | 1583

**C#.NET 8.0**

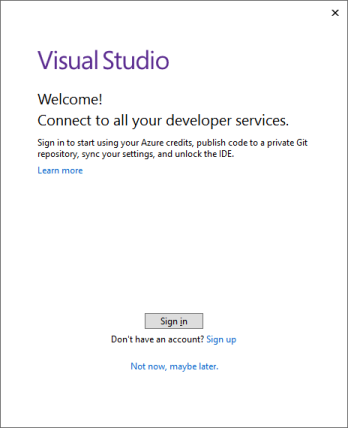
****

If you have Microsoft account already and click on “Sign in” and complete the login process. If you don’t have Microsoft account, click on “Sign up” and complete the registration process.

Note: If you don’t want to login, click on “Not now, may be later”; but then Visual Studio expires in 30 days.

D. Harsha Vardhan (.NET Expert) P a g e 45 | 1583

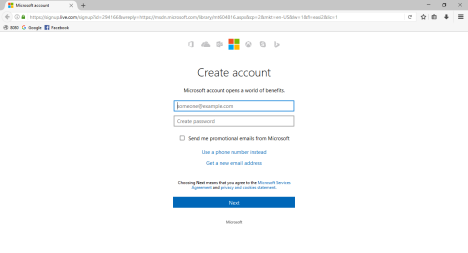
**C#.NET 8.0**

****

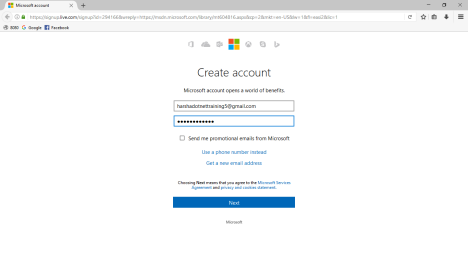
If you click on “Sign up”, you will get the following page.

D. Harsha Vardhan (.NET Expert) P a g e 46 | 1583

**C#.NET 8.0**

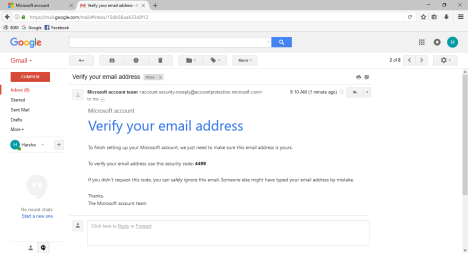
****

Enter your email id and enter new password.

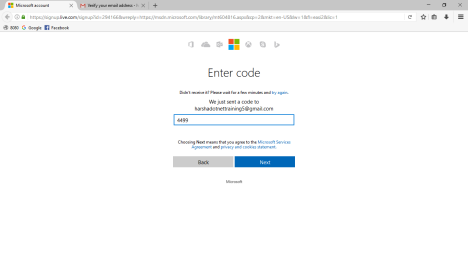
Click on “Next”.

D. Harsha Vardhan (.NET Expert) P a g e 47 | 1583

**C#.NET 8.0**

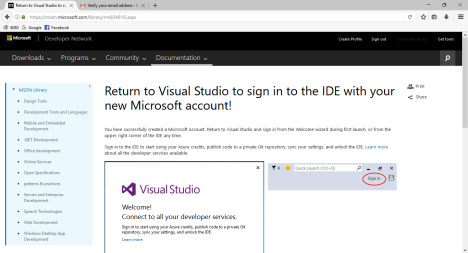
****

You will get a security code to your email. Login into your gmail, check the code that you have received and enter it in this page.

After entering the security code, click on “Next”.

D. Harsha Vardhan (.NET Expert) P a g e 48 | 1583

**C#.NET 8.0**

****

If you get the above page, you have successfully registered.

D. Harsha Vardhan (.NET Expert) P a g e 49 | 1583

**C#.NET 8.0**

Now click on “Sign in” in Visual Studio and enter the email.

Enter the password.

D. Harsha Vardhan (.NET Expert) P a g e 50 | 1583

**C#.NET 8.0**

****

Click on “Sign in”.

D. Harsha Vardhan (.NET Expert) P a g e 51 | 1583

**C#.NET 8.0**

****

D. Harsha Vardhan (.NET Expert) P a g e 52 | 1583

**C#.NET 8.0**

****

Select “Development Settings” as “General”. Select “color theme” as “Blue”.

Click on “Start Visual Studio”.

D. Harsha Vardhan (.NET Expert) P a g e 53 | 1583

**C#.NET 8.0**

****

Visual Studio Installation has been completed successfully.

D. Harsha Vardhan (.NET Expert) P a g e 54 | 1583

**C#.NET 8.0**

2

**C#.NET – Language**

**Fundamentals**

Introduction to C#.NET

What is C#.NET:

C#.NET is the .NET’s most popular programming language, which is used to create stand-alone applications (console applications and windows applications, windows services) primarily.

• C#.NET is developed on basis of C, C++.

• C#.NET is an “Object Oriented Programming Language”.

• C#.NET is a case sensitive language.

• C#.NET is an advanced and matured language.

• C#.NET is a high-level programming language.

• C#.NET is a compiler-based language.

• C#.NET is a part of .NET Framework.

• C#.NET programs run based on the CLR (.NET run time environment).

D. Harsha Vardhan (.NET Expert) P a g e 55 | 1583

**C#.NET 8.0**

Versions of C#.NET

| Sl. No | C#.NET Version | .NET Framework Version | Year of release |
| --- | --- | --- | --- |
| 1 | C#.NET 1.0 | .NET Framework 1.0 | 2002 |
| 2 | C#.NET 1.1 | .NET Framework 1.1 | 2003 |
| 3 | C#.NET 2.0 | .NET Framework 2.0 | 2005 |
| 4 | C#.NET 3.0 | .NET Framework 3.0 | 2006 |
| 5 | C#.NET 3.5 | .NET Framework 3.5 | 2007 |
| 6 | C#.NET 4.0 | .NET Framework 4.0 | 2010 |
| 7 | C#.NET 5.0 | .NET Framework 4.5 | 2012 |
| 8 | C#.NET 5.1 | .NET Framework 4.5.1 | 2013 |
| 9 | C#.NET 5.2 | .NET Framework 4.5.2 | 2014 |
| 10 | C#.NET 6.0 | .NET Framework 4.6 | 2015 |
| 11 | C#.NET 7.0 | .NET Framework 4.7 | 2017 |
| 12 | C#.NET 8.0 | .NET Framework 4.8 | 2019 |

Tokens of C#.NET

• In "C#.NET", we have to use the following tokens (parts of the language).

| Sl.  No | Token | Description |
| --- | --- | --- |
| 1 | Keywords | abstract, as, base, bool, break, byte, case, catch, char, class, const, continue, decimal, default, delegate, do, double, else, enum, event, false, finally, float, for, foreach, goto, if, in, int, interface, internal, is, long, namespace, new, null, object, out, override, private, protected, public, readonly, ref, return, sbyte, sealed, short, sizeof, static, string, struct, switch, this, throw, true, try, typeof, uint, ulong, ushort, using, virtual, void, while, async, await, from, join, let, orderby, partial, set, get, value, var, where |
| 2 | Operators | Any symbols such as +, -, \*, /, %, =, == etc. |

D. Harsha Vardhan (.NET Expert) P a g e 56 | 1583

**C#.NET 8.0**

| 3 | Literals | Fixed values are literals.  • Integer literals: Any number without decimal part. Ex: 10 • Floating-point literals: Any number with decimal part. Ex: 10.87 • Character literals: Any single character in single quotes. Ex: ‘A’.  The character can be alphabet, digit, space or special symbol. • String literals: One or more characters in double quotes. Ex: “Abc 123$”  • Boolean literals: true / false |
| --- | --- | --- |
| 4 | Identifiers | Any user-defined names: abc |

Naming Conventions for Identifiers

• Class names, Interface names, Namespace names, Structure names, Method names, Property names: PascalCasing

• Local variables: camelCasing

• Public Fields: PascalCasing

• Private Fields: camelCasing with underscore ( \_ )

Identifier Naming Rules

You must follow the below rules while giving name for identifier.

• Identifier names can’t be same as keyword.

• Identifier names should not have spaces.

• Identifier names should not have special characters.

• Duplicate identifier names not allowed.

D. Harsha Vardhan (.NET Expert) P a g e 57 | 1583

**C#.NET 8.0**

Console Applications - Introduction

What is Console Application?

• Console applications are the programs that run on “Command Prompt” window. • Console applications support CUI (Character User Interface).

• In console applications, all the input and output will be in the form of characters only. • Console applications are not user-friendly.

• Console applications are not used in real time, but good for learning programming basics and OOP concepts.

Rules of Console Application

• Console application must have atleast one class; which is called as “Main class”. • The “Main class” must contain atleast one method; which is called as “Main method”. • The “main method” name should be “Main”; “M” is capital letter.

• When the program execution starts the “Main” method will be automatically executed. Main method is the “starting point of the program execution”.

• The Main method should be “static” method, because it should be called without creating any object for the class.

Syntax:

**class classname**

**{**

**static void Main()**

**{**

**}**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 58 | 1583

**C#.NET 8.0**

Console Application - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “Project1”.

• Type the location as “C:\CSharp”.

• Type the solution name as “Solution1”.

• Click on OK.

• It shows “Program.cs” file automatically.

• Type the following code for “Program.cs” file:

**Program.cs**

**class Sample**

**{**

**static void Main()**

**{**

**System.Console.Write("Hello World"); }**

**}**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

D. Harsha Vardhan (.NET Expert) P a g e 59 | 1583

**C#.NET 8.0**

**Output**

****

The “System.Console” class

**The “System.Console” class**

• C#.NET provides “System.Console” class to perform I/O operations in console applications. • “System” is a namespace; “Console” is a class.

• The following are the important methods of “System.Console” class.

1. System.Console.Write()

2. System.Console.WriteLine()

3. System.Console.ReadKey()

4. System.Console.ReadLine()

5. System.Console.Clear()

1. System.Console.Write()

• This statement is used to display the given value on the command prompt window. • After printing the value, the cursor will be kept in the same line.

• Syntax: System.Console.Write(value);

• Example: System.Console.Write(100);

D. Harsha Vardhan (.NET Expert) P a g e 60 | 1583

**C#.NET 8.0**

2. System.Console.WriteLine()

• This statement is used to display the given value on the command prompt window. • After printing the value, the cursor will be automatically moved to the next line. • Syntax: System.Console.WriteLine(value);

• Example: System.Console.WriteLine(100);

3. System.Console.ReadKey()

• This statement is used to wait until the user presses any key on the keyboard. • Syntax: System.Console.ReadKey();

• Example: System.Console.ReadKey();

4. System.Console.ReadLine()

• This statement is used to accept a string value from keyboard.

• Syntax: System.Console.ReadLine();

• Example: System.Console.ReadLine();

5. System.Console.Clear()

• This statement is used to clear the screen.

• Syntax: System.Console.Clear();

• Example: System.Console.Clear();

D. Harsha Vardhan (.NET Expert) P a g e 61 | 1583

**C#.NET 8.0**

System.Console.WriteLine – Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “WriteLineExample”.

• Type the location as “C:\CSharp”.

• Type the solution name as “WriteLineExample”. • Click on OK.

• It shows “Program.cs” file automatically.

• Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**System.Console.WriteLine("Hello"); System.Console.WriteLine("Hello"); System.Console.WriteLine("Hello"); System.Console.ReadKey();**

**}**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 62 | 1583

**C#.NET 8.0**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

**Output**

****

System.Console.ReadKey – Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “ReadKeyExample”.

• Type the location as “C:\CSharp”.

• Type the solution name as “ReadKeyExample”.

• Click on OK.

• It shows “Program.cs” file automatically.

D. Harsha Vardhan (.NET Expert) P a g e 63 | 1583

**C#.NET 8.0**

• Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//display message**

**System.Console.WriteLine("Hello");**

**//wait for pressing any key on the keyboard System.Console.ReadKey();**

**//display message**

**System.Console.WriteLine("Hello");**

**//wait for pressing any key on the keyboard System.Console.ReadKey();**

**//display message**

**System.Console.WriteLine("Hello");**

**//wait for pressing any key on the keyboard System.Console.ReadKey();**

**}**

**}**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

D. Harsha Vardhan (.NET Expert) P a g e 64 | 1583

**C#.NET 8.0**

**Output**

****Press Enter.

Press Enter.



System.Console.Clear - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “ClearExample”.

D. Harsha Vardhan (.NET Expert) P a g e 65 | 1583

**C#.NET 8.0**

• Type the location as “C:\CSharp”.

• Type the solution name as “ClearExample”.

• Click on OK.

• It shows “Program.cs” file automatically.

• Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//display message**

**System.Console.WriteLine("Hello"); //wait for pressing any key on the keyboard System.Console.ReadKey();**

**//clear the screen**

**System.Console.Clear();**

**//display message**

**System.Console.WriteLine("how"); //wait for pressing any key on the keyboard System.Console.ReadKey();**

**//clear the screen**

**System.Console.Clear();**

**//display message**

**System.Console.WriteLine("are you"); //wait for pressing any key on the keyboard System.Console.ReadKey();**

**}**

**}**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

D. Harsha Vardhan (.NET Expert) P a g e 66 | 1583

**Output**

**C#.NET 8.0**

****Press Enter.

Press Enter.



D. Harsha Vardhan (.NET Expert) P a g e 67 | 1583

**C#.NET 8.0**

Variables

**Variables**

• A variable is a named memory location in RAM, to store a particular type of value temporarily while the program is running.

• All the variables stored in RAM (temporary memory).

• All the variables must be declared before its usage. While declaring variables, data type is to be specified. Based on the data type, the amount of memory to be allocated will be decided. Once a variable is declared, we can’t change the “variable name” or “variable’s data type”.

• The variables memory will be allocated when the program execution starts; and all the variables will be deleted (de-allocated) from memory automatically, at the end of the program.

• A variable can store only one value. If you assign another value, the old value will be overwritten.

• We can change the value of a variable any no. of times.

• Syntax to create a variable:

datatype variablename ;

• Syntax to set value into a variable:

variablename = value ;

• Syntax to get the value of a variable:

variablename

D. Harsha Vardhan (.NET Expert) P a g e 68 | 1583

**C#.NET 8.0**

Variables - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “VariablesExample”. • Type the location as “C:\CSharp”.

• Type the solution name as “VariablesExample”. • Click on OK.

• It shows “Program.cs” file automatically.

• Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//create a variable**

**int x;**

**//set value into the variable**

**x = 10;**

**//get the value of variable**

**System.Console.WriteLine(x);**

**//wait for pressing any key on the keyboard System.Console.ReadKey();**

**}**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 69 | 1583

**C#.NET 8.0**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

**Output**

****

D. Harsha Vardhan (.NET Expert) P a g e 70 | 1583

**C#.NET 8.0**

Data Types

**Data Type**

• A data type is a concept, which specifies the type of the data that is to be stored in a variable.

Types of Data types

• Data types are two types:

I. Numerical data types

II. Non-numerical data types

Numerical Data Types

| Sl.  No | Data  Type | Description | No. of  Bytes | Range | Suitable to  store |
| --- | --- | --- | --- | --- | --- |
| 1 | sbyte | 8-bit signed integer | 1 byte | -128 to 127 | Very small  positive or  negative  integers |
| 2 | byte | 8-bit  unsigned  integer | 1 byte | 0 to 255 | Very small  positive  integers |
| 3 | short | 16-bit signed integer | 2 bytes | -32,768 to 32,767 | Small positive or negative  integers |
| 4 | ushort | 16-bit  unsigned  integer | 2 bytes | 0 to 65,535 | Small positive integers |

D. Harsha Vardhan (.NET Expert) P a g e 71 | 1583

**C#.NET 8.0**

| 5 | int | 32-bit signed integer | 4 bytes | -2,147,483,648 to 2,147,483,647 | Medium  positive or  negative  integers |
| --- | --- | --- | --- | --- | --- |
| 6 | uint | 32-bit  unsigned  integer | 4 bytes | 0 to 4,294,967,295 | Medium  positive  integers |
| 7 | long | 64-bit signed integer | 8 bytes | -9,223,372,036,854,775,808 to  9,223,372,036,854,770,000 | Large positive or  negative  integers |
| 8 | ulong | 64-bit  unsigned  integer | 8 bytes | 0 to 18,446,744,073,709,551,615 | Large positive integers |
| 9 | float | Signed  floating  point  number | 4 bytes | -3.402823E+38 to 3.402823E+38 (Precision: 7 digits) | Small  floating-point numbers |
| 10 | double | Signed  floating  point  number | 8 bytes | -1.79769313486232E+308 to  1.79769313486232E+308  (Precision: 15 digits) | Medium  floating-point numbers |
| 11 | decimal | Signed  floating  point  number | 16  bytes | -79228162514264337593543950335 to 79228162514264337593543950335 (Precision: 28 digits) | Large  floating-point numbers |

Default Numerical data types

• C# compiler automatically treats a “number without decimal part” as “int” data type, if it is within the maximum limit of “int” data type.

D. Harsha Vardhan (.NET Expert) P a g e 72 | 1583

**C#.NET 8.0**

• C# compiler automatically treats a “number without decimal part” as “long” data type, if it exceeds the limit of “int” data type.

• C# compiler automatically treats a number with decimal part as “double” data type.

Non-Numerical Data Types

| Sl.  No | Data Type | Description | Value format | No. of Bytes |
| --- | --- | --- | --- | --- |
| 1 | char | To store single character | ‘character’ | 2 bytes |
| 2 | string | To store one or more  characters.  Max: 2 billion characters | "string here" | No. of characters \* 2 |
| 3 | bool | To store true / false values. | true or false | 1 bit |

Numerical Data Types - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “NumericalDataTypesExample”.

• Type the location as “C:\CSharp”.

D. Harsha Vardhan (.NET Expert) P a g e 73 | 1583

**C#.NET 8.0**

• Type the solution name as “NumericalDataTypesExample”. • Click on OK.

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//create variables for all numerical data types sbyte a = 10;**

**byte b = 20;**

**short c = 30;**

**ushort d = 40;**

**int e = 50;**

**uint f = 60;**

**long g = 70;**

**ulong h = 80;**

**float i = 90.23F;**

**double j = 100.23489;**

**decimal k = 110.882932M;**

**//displays the values of all variables System.Console.WriteLine(a);**

**System.Console.WriteLine(b);**

**System.Console.WriteLine(c);**

**System.Console.WriteLine(d);**

**System.Console.WriteLine(e);**

**System.Console.WriteLine(f);**

**System.Console.WriteLine(g);**

**System.Console.WriteLine(h);**

**System.Console.WriteLine(i);**

**System.Console.WriteLine(j);**

**System.Console.WriteLine(k);**

**System.Console.ReadKey();**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 74 | 1583

**C#.NET 8.0**

**}**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

**Output**

****

D. Harsha Vardhan (.NET Expert) P a g e 75 | 1583

**C#.NET 8.0**

MinValue and MaxValue

**NumericalDataType.MinValue**

• This statement returns the minimum value of the specified numerical data type. • Syntax: datatype.MinValue

• Ex: int.MinValue

**NumericalDataType.MaxValue**

• This statement returns the maximum value of the specified numerical data type. • Syntax: datatype.MaxValue

• Ex: int.MaxValue

MinValue and MaxValue - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “MinValueMaxValueExample”.

• Type the location as “C:\CSharp”.

• Type the solution name as “MinValueMaxValueExample”.

D. Harsha Vardhan (.NET Expert) P a g e 76 | 1583

**C#.NET 8.0**

• Click on OK.

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//get the minimum value of "int" data type**

**int min = int.MinValue;**

**//get the maximum value of "int" data type**

**int max = int.MaxValue;**

**//display the min and max values**

**System.Console.WriteLine(min);**

**System.Console.WriteLine(max);**

**System.Console.ReadKey();**

**}**

**}**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

**Output**

****

D. Harsha Vardhan (.NET Expert) P a g e 77 | 1583

**C#.NET 8.0**

char - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”.

• Select “Visual C#”.

• Select “Console Application”.

• Type the project name as “charExample”. • Type the location as “C:\CSharp”.

• Type the solution name as “charExample”. • Click on OK.

• It shows “Program.cs” file automatically. • Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//create variable of "char" data type char ch = 'A';**

**//display the value of the variable System.Console.WriteLine(ch); System.Console.ReadKey(); }**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 78 | 1583

**C#.NET 8.0**

**Running the Project**

• Go to “Debug” menu and click on “Start Debugging”.

**Output**

****

String

**String**

• String: A string is a group of characters.

• String literal should be written inside the double quotes.

• All the names are strings. Ex: person names, city names, country names etc. • String may contain alphabets, numbers, spaces and also special symbols.

• Alpha-numerical values are also treated as strings. Ex: car number, phone number, bank ifsc codes etc.

• Syntax to create a string variable:

o string variablename = "value";

D. Harsha Vardhan (.NET Expert) P a g e 79 | 1583

**C#.NET 8.0**

String - Example

**Creating Project**

• Open Visual Studio 2019.

• Go to “File” – “New” – “Project”.

• Select “.NET Framework 4.8”. Select “Visual C#”. • Select “Console Application”.

• Type the project name as “StringExample”.

• Type the location as “C:\CSharp”.

• Type the solution name as “StringExample”. Click on OK. • It shows “Program.cs” file automatically.

• Type the following code for “Program.cs” file:

**Program.cs**

**class Program**

**{**

**static void Main()**

**{**

**//create a variable**

**string s = "Hello 123 $#&";**

**//display the value of the variable System.Console.WriteLine(s);**

**System.Console.WriteLine(s);**

**System.Console.WriteLine(s);**

**System.Console.ReadKey();**

**}**

D. Harsha Vardhan (.NET Expert) P a g e 80 | 1583