Dhruvil Dobariya

dhruvildobariya21@gmail.com

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

GUI Basics

Documentation



Index

[1 Visual Studio 2019 IDE Overview 1](#_Toc120095063)

[1.1 What is a Visual Studio? 1](#_Toc120095064)

[1.2 Different types of windows 1](#_Toc120095065)

[1.3 Solution and Project 2](#_Toc120095066)

[1.4 Code editor features 4](#_Toc120095067)

[1.5 Popular keyboard shortcuts for Visual Studio: 9](#_Toc120095068)

[2 Project Types 17](#_Toc120095069)

[2.1 Windows Development 17](#_Toc120095070)

[2.2 Class Library 18](#_Toc120095071)

[2.3 Mobile Development 19](#_Toc120095072)

[2.4 Web Development 20](#_Toc120095073)

[3 Introduction To C# 22](#_Toc120095074)

[3.1 What is a C#? 22](#_Toc120095075)

[3.2 “Hello World” Program 22](#_Toc120095076)

[3.3 Namespace: 23](#_Toc120095077)

[3.4 The “using” Keyword 24](#_Toc120095078)

[3.5 Class 25](#_Toc120095079)

[3.6 Methods 26](#_Toc120095080)

[3.7 Variables 27](#_Toc120095081)

[4 Understanding C# Program 28](#_Toc120095082)

[4.1 Program flow 28](#_Toc120095083)

[4.2 Understanding Syntax 28](#_Toc120095084)

[4.3 Common Language Runtime (CLR) 29](#_Toc120095085)

[5 Working with code files, projects & solutions 30](#_Toc120095086)

[5.1 Structure of solution 30](#_Toc120095087)

[5.2 Understanding structure of project 30](#_Toc120095088)

[6 Understanding datatypes & variables with conversion 37](#_Toc120095089)

[6.1 Data Type: 37](#_Toc120095090)

[6.2 Datatype Conversion 41](#_Toc120095091)

[7 Understanding Decision making & statements 46](#_Toc120095092)

[7.1 If Else 46](#_Toc120095093)

[7.2 Switch 53](#_Toc120095094)

[8 Operators and Expressions 57](#_Toc120095095)

[8.1 Operators 57](#_Toc120095096)

[9 Loop Iteration 62](#_Toc120095097)

[9.1 While 62](#_Toc120095098)

[9.2 Do While 63](#_Toc120095099)

[9.3 For 65](#_Toc120095100)

[9.4 For Each 66](#_Toc120095101)

[9.5 Break Statement 68](#_Toc120095102)

[9.6 Continue Statement 69](#_Toc120095103)

[9.7 Go To Statement 71](#_Toc120095104)

[10 Understanding Arrays 72](#_Toc120095105)

[10.1 Introduction 72](#_Toc120095106)

[10.2 Declaring Array 72](#_Toc120095107)

[10.3 Single dimensional array 73](#_Toc120095108)

[10.4 Multidimensional array 74](#_Toc120095109)

[10.5 Jugged array 77](#_Toc120095110)

[10.6 Array Class 79](#_Toc120095111)

[11 Defining and Calling Methods 83](#_Toc120095112)

[11.1 Introduction 83](#_Toc120095113)

[11.2 Defining the method 83](#_Toc120095114)

[11.3 Calling Method 84](#_Toc120095115)

[11.4 Passing Parameter to the method 85](#_Toc120095116)

[12 Working with strings 86](#_Toc120095117)

[12.1 Introduction 86](#_Toc120095118)

[12.2 String 86](#_Toc120095119)

[12.3 StringBuilder 87](#_Toc120095120)

[13 Working with Datetimes 90](#_Toc120095121)

[13.1 Introduction 90](#_Toc120095122)

[13.2 DateTime 90](#_Toc120095123)

[14 Understanding Classes 100](#_Toc120095124)

[14.1 Introduction 100](#_Toc120095125)

[14.2 Types of Classes 101](#_Toc120095126)

[15 Depth in Classes 104](#_Toc120095127)

[15.1 Object 104](#_Toc120095128)

[16 Scope & Accessibility Modifiers 106](#_Toc120095129)

[16.1 Introduction 106](#_Toc120095130)

[16.2 Access Specifiers 106](#_Toc120095131)

[17 Namespace & .Net Library 107](#_Toc120095132)

[17.1 Introduction 107](#_Toc120095133)

[17.2 Full Name 107](#_Toc120095134)

[17.3 With “Using” Keyword 107](#_Toc120095135)

[18 Creating and adding ref. to assemblies 109](#_Toc120095136)

[18.1 Introduction 109](#_Toc120095137)

[18.2 Create By Self 109](#_Toc120095138)

[18.3 Provide By .NET 110](#_Toc120095139)

[18.4 Provide By third-party 110](#_Toc120095140)

[19 Working with collections 111](#_Toc120095141)

[19.1 Introduction 111](#_Toc120095142)

[19.2 System.Collection.Genaric Class 111](#_Toc120095143)

[19.3 System.Collection Class 113](#_Toc120095144)

[19.4 System.Collection.Concurrent 123](#_Toc120095145)

[20 Enumerations 125](#_Toc120095146)

[20.1 Enum: 125](#_Toc120095147)

[21 Exception Handling 127](#_Toc120095148)

[21.1 Introduction 127](#_Toc120095149)

[21.2 Exception Class 128](#_Toc120095150)

[22 Events 130](#_Toc120095151)

[22.1 Introduction 130](#_Toc120095152)

[22.2 Implement Event 131](#_Toc120095153)

[22.3 Delegates 131](#_Toc120095154)

[23 Basics of File Handling 133](#_Toc120095155)

[23.1 Introduction 133](#_Toc120095156)

[23.2 FileSteram Class 133](#_Toc120095157)

[23.3 StreamWriter Class 134](#_Toc120095158)

[23.4 SteamReader Class 135](#_Toc120095159)

[23.5 BinaryWriter Class 136](#_Toc120095160)

[23.6 BinaryRender Class 138](#_Toc120095161)

[23.7 DirectoryInfo Class 139](#_Toc120095162)

[23.8 FileInfo Class 140](#_Toc120095163)

[24 Interface & inheritance 143](#_Toc120095164)

[24.1 Inheritance 143](#_Toc120095165)

[24.2 Interface 147](#_Toc120095166)

[25 Introduction to WEB Development 148](#_Toc120095167)

[25.1 ASP.NET Web Forms 148](#_Toc120095168)

[25.2 ASP.NET MVC 158](#_Toc120095169)

[25.3 ASP.NET Rest Web API 159](#_Toc120095170)

1. Visual Studio 2019 IDE Overview

# What is a Visual Studio?

* Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
* Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists.

# Different types of windows



* In Solution Explorer, at upper right, you can view, navigate, and manage your code files. Solution Explorer can help organize your code by grouping the files into solutions and projects.
* The central editor window, where you'll probably spend most of your time, displays file contents. In the editor window, you can edit code or design a user interface such as a window with buttons and text boxes.
* The central editor window, where you'll probably spend most of your time, displays file contents. In the editor window, you can edit code or design a user interface such as a window with buttons and text boxes.

# Solution and Project

* We are used Solution Explorer to manage project. Using Solution Explorer we should manage folder structure and navigate different files.
* Solution Explorer have a file which is contain solution details and projects details.
* This file have *‘.sln’* extension.
* Solution contain one or more projects.
* Project have one file which containe details about project.
* Which have *‘.csproj’* extension(if project base on C#).

## Create a solution:

* Open Visual Studio, and on the start window, select Create a new project.
* On the Configure your new project page, give the name of Solution, and then select Create.
* Let say our solution name is **'QuickSolution'**.

## Add a project:

* Right-click Solution **'QuickSolution'** in Solution Explorer, and select Add > New Project from the context menu.
* On the Add a new project page, type empty into the search box at the top, and select C# under All languages.
* Select the C# Empty Project (.NET Framework) template, and then select Next.
* On the Configure your new project page, give the name of Project, and then select Create.
* Let say our solution name is **‘QuickDate’**.

## Add an item to the project:

* From the right-click or context menu of the **‘QuickDate’** project in Solution Explorer, select Add > New Item.
* Expand Visual C# Items, and then select Code. In the middle pane, select the Class item template. Under Name, give the name of file, and then select Add.
* Let say our file name is **‘Calander.cs’**.

## Add a second project:

* From the right-click or context menu of Solution '**QuickSolution'** in Solution Explorer, select Add > New Project.
* In the Add a new project dialog box, type unit test into the search box at the top, and then select C# under All languages.
* Select the C# Unit Test Project (.NET Framework) project template, and then select Next.
* On the Configure your new project page, give the name of project, and then select Create.
* Let say our second project name is **‘QuickTest’.**
* Add one file inside the second project.
* Let say our file name is **‘UnitTest1.cs’**.

## Add a project reference:

* In Solution Explorer, right-click the References node of the **‘QuickTest’** project, and select Add Reference from the context menu.
* In the Reference Manager dialog box, under Projects, select the checkbox next to **‘QuickDate’,** and then select OK.
* Now a reference to the **‘QuickDate’** project appears under the **‘QuickTest’** project in Solution Explorer.

## Project properties:

* In Solution Explorer, right-click the **‘QuickTest’** project and select Properties, or select the project and press Alt+Enter.

# Code editor features

* Some popular features in Visual Studio that improve your productivity when developing software include

## Squiggles and Quick Actions:

* Squiggles are wavy underlines that alert you to errors or potential problems in your code as you type.
* These visual clues help you fix problems immediately, without waiting to discover errors during build or runtime.
* If you hover over a squiggle, you see more information about the error.
* A lightbulb might also appear in the left margin showing Quick Actions you can take to fix the error.

## Code Cleanup:

* With the click of a button, you can format your code and apply any code fixes suggested by your code style settings, .editorconfig conventions, and Roslyn analyzers.
* Code Cleanup, currently available for C# code only, helps you resolve issues in your code before it goes to code review.



## Refactoring:

* Refactoring includes operations such as intelligent renaming of variables, extracting one or more lines of code into a new method, and changing the order of method parameters.



## IntelliSense:

* IntelliSense is a set of features that display information about your code directly in the edor and, in some cases, write small bits of code for you.
* It's like having basic documentation inline in the editor, so you don't have to look up type information elsewhere.

## Visual Studio search:

* Visual Studio menus, options, and properties can seem overwhelming at times.
* Visual Studio search, or *‘Ctrl+Q’*, is a great way to rapidly find IDE features and code in one place.

Graphical user interface, text

Description automatically generated

## Live Share:

* Collaboratively edit and debug with others in real time, regardless of your app type or programming language.
* You can instantly and securely share your project. You can also share debugging sessions, terminal instances, localhost web apps, voice calls, and more.

## Call Hierarchy:

* The Call Hierarchy window shows the methods that call a selected method.
* This information can be useful when you're thinking about changing or removing the method, or when you're trying to track down a bug.



## CodeLens:

* CodeLens helps you find code references, code changes, linked bugs, work items, code reviews, and unit tests, without leaving the editor.



## Go To Definition:

* The Go To Definition feature takes you directly to the location of a function or type definition.



## Peek Definition:

* The Peek Definition window shows a method or type definition without opening a separate file.

# Popular keyboard shortcuts for Visual Studio:

## Build:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts | Command ID |
| Build solution | Ctrl+Shift+B | Build.BuildSolution |
| Cancel | Ctrl+Break | Build.Cancel |
| Compile | Ctrl+F7 | Build.Compile |
| Run code analysis on solution | Alt+F11 | Build.RunCodeAnalysisonSolution |

## Debug:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Break at function | Ctrl+B | Debug.BreakatFunction |
| Break all | Ctrl+Alt+Break | Debug.BreakAll |
| Delete all breakpoints | Ctrl+Shift+F9 | Debug.DeleteAllBreakpoints |
| Exceptions | Ctrl+Alt+E | Debug.Exceptions |
| Quick watch | Ctrl+Alt+Q  or Shift+F9 | Debug.QuickWatch |
| Restart | Ctrl+Shift+F5 | Debug.Restart |
| Run to cursor | Ctrl+F10 | Debug.RunToCursor |
| Set next statement | Ctrl+Shift+F10 | Debug.SetNextStatement |
| Start | F5 | Debug.Start |
| Start without debugging | Ctrl+F5 | Debug.StartWithoutDebugging |
| Step into | F11 | Debug.StepInto |
| Step out | Shift+F11 | Debug.StepOut |
| Step over | F10 | Debug.StepOver |
| Stop debugging | Shift+F5 | Debug.StopDebugging |
| Toggle breakpoint | F9 | Debug.ToggleBreakpoint |

## Edit:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Break line | Enter [Text Editor, Report Designer, Windows Forms Designer]  or Shift+Enter [Text Editor] | Edit.BreakLine |
| Collapse to definitions | Ctrl+M, Ctrl+O [Text Editor] | Edit.CollapseToDefinitions |
| Comment selection | Ctrl+K, Ctrl+C [Text Editor] | Edit.CommentSelection |
| Complete word | Alt+Right Arrow [Text Editor, Workflow Designer]  or Ctrl+Spacebar [Text Editor, Workflow Designer]  or Ctrl+K, W [Workflow Designer]  or Ctrl+K, Ctrl+W [Workflow Designer] | Edit.CompleteWord |
| Copy | Ctrl+C  or Ctrl+Insert | Edit.Copy |
| Cut | Ctrl+X  or Shift+Delete | Edit.Cut |
| Delete | Delete [Team Explorer]  or Shift+Delete [Sequence Diagram, UML Activity Diagram, Layer Diagram]  or Ctrl+Delete [Class Diagram] | Edit.Delete |
| Find | Ctrl+F | Edit.Find |
| Find all references | Shift+F12 | Edit.FindAllReferences |
| Find in files | Ctrl+Shift+F | Edit.FindinFiles |
| Find next | F3 | Edit.FindNext |
| Find next selected | Ctrl+F3 | Edit.FindNextSelected |
| Format document | Ctrl+K, Ctrl+D [Text Editor] | Edit.FormatDocument |
| Format selection | Ctrl+K, Ctrl+F [Text Editor] | Edit.FormatSelection |
| Go to | Ctrl+G | Edit.GoTo |
| Go to declaration | Ctrl+F12 | Edit.GoToDeclaration |
| Go to definition | F12 | Edit.GoToDefinition |
| Go to find combo | Ctrl+D | Edit.GoToFindCombo |
| Go to next location | F8 | Edit.GoToNextLocation |
| Insert snippet | Ctrl+K, Ctrl+X | Edit.InsertSnippet |
| Insert tab | Tab [Report Designer, Windows Forms Designer, Text Editor] | Edit.InsertTab |
| Line cut | Ctrl+L [Text Editor] | Edit.LineCut |
| Line down extend column | Shift+Alt+Down Arrow [Text Editor] | Edit.LineDownExtendColumn |
| Line open above | Ctrl+Enter [Text Editor] | Edit.LineOpenAbove |
| List members | Ctrl+J [Text Editor, Workflow Designer]  or Ctrl+K, Ctrl+L [Workflow Designer]  or Ctrl+K, L [Workflow Designer] | Edit.ListMembers |
| Navigate to | Ctrl+, | Edit.NavigateTo |
| Open file | Ctrl+Shift+G | Edit.OpenFile |
| Overtype mode | Insert [Text Editor] | Edit.OvertypeMode |
| Parameter info | Ctrl+Shift+Spacebar [Text Editor, Workflow Designer] or Ctrl+K, Ctrl+P [Workflow Designer] or Ctrl+K, P [Workflow Designer] | Edit.ParameterInfo |
| Paste | Ctrl+V or Shift+Insert | Edit.Paste |
| Peek definition | Alt+F12 [Text Editor] | Edit.PeekDefinition |
| Redo | Ctrl+Y or Shift+Alt+Backspace or Ctrl+Shift+Z | Edit.Redo |
| Replace | Ctrl+H | Edit.Replace |
| Select all | Ctrl+A | Edit.SelectAll |
| Select current word | Ctrl+W [Text Editor] | Edit.SelectCurrentWord |
| Selection cancel | Esc [Text Editor, Report Designer, Settings Designer, Windows Forms Designer, Managed Resources Editor] | Edit.SelectionCancel |
| Surround with | Ctrl+K, Ctrl+S (available only in Visual Studio 2019 and earlier) | Edit.SurroundWith |
| Tab left | Shift+Tab [Text Editor, Report Designer, Windows Forms Editor] | Edit.TabLeft |
| Toggle all outlining | Ctrl+M, Ctrl+L [Text Editor] | Edit.ToggleAllOutlining |
| Toggle bookmark | Ctrl+K, Ctrl+K [Text Editor] | Edit.ToggleBookmark |
| Toggle completion mode | Ctrl+Alt+Space [Text Editor] | Edit.ToggleCompletionMode |
| Toggle outlining expansion | Ctrl+M, Ctrl+M [Text Editor] | Edit.ToggleOutliningExpansion |
| Uncomment selection | Ctrl+K, Ctrl+U [Text Editor] | Edit.UncommentSelection |
| Undo | Ctrl+Z or Alt+Backspace | Edit.Undo |
| Word delete to end | Ctrl+Delete [Text Editor] | Edit.WordDeleteToEnd |
| Word delete to start | Ctrl+Backspace [Text Editor] | Edit.WordDeleteToStart |

## File: popular shortcuts:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Exit | Alt+F4 | File.Exit |
| New file | Ctrl+N | File.NewFile |
| New project | Ctrl+Shift+N | File.NewProject |
| New web site | Shift+Alt+N | File.NewWebSite |
| Open file | Ctrl+O | File.OpenFile |
| Open project | Ctrl+Shift+O | File.OpenProject |
| Open web site | Shift+Alt+O | File.OpenWebSite |
| Rename | F2 [Team Explorer] | File.Rename |
| Save all | Ctrl+Shift+S | File.SaveAll |
| Save selected items | Ctrl+S | File.SaveSelectedItems |
| View in browser | Ctrl+Shift+W | File.ViewinBrowser |

## Project:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Add existing item | Shift+Alt+A | Project.AddExistingItem |
| Add new item | Ctrl+Shift+A | Project.AddNewItem |

## Refactor:

|  |  |  |
| --- | --- | --- |
| Command | Keyboard shortcut [Special contexts] | Command ID |
| Extract method | Ctrl+R, Ctrl+M | Refactor.ExtractMethod |

## Tools:

|  |  |  |
| --- | --- | --- |
| Command | Keyboard shortcut [Special contexts] | Command ID |
| Attach to process | Ctrl+Alt+P | Tools.AttachtoProcess |

## View:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Class view | Ctrl+Shift+C | View.ClassView |
| Edit label | F2 | View.EditLabel |
| Error list | Ctrl+\, Ctrl+E or Ctrl+\, E | View.ErrorList |
| Navigate backward | Ctrl+- | View.NavigateBackward |
| Navigate forward | Ctrl+Shift+- | View.NavigateForward |
| Object browser | Ctrl+Alt+J | View.ObjectBrowser |
| Output | Ctrl+Alt+O | View.Output |
| Properties window | F4 | View.PropertiesWindow |
| Refresh | F5 [Team Explorer] | View.Refresh |
| Server explorer | Ctrl+Alt+S | View.ServerExplorer |
| Show smart tag | Ctrl+.  or Shift+Alt+F10 [HTML Editor Design View] | View.ShowSmartTag |
| Solution explorer | Ctrl+Alt+L | View.SolutionExplorer |
| TFS Team Explorer | Ctrl+\, Ctrl+M | View.TfsTeamExplorer |
| Toolbox | Ctrl+Alt+X | View.Toolbox |
| View code | Enter [Class Diagram]  or F7 [Settings Designer] | View.ViewCode |
| View designer | Shift+F7 [HTML Editor Source View] | View.ViewDesigner |

## Window:

|  |  |  |
| --- | --- | --- |
| Commands | Keyboard shortcuts [Special contexts] | Command ID |
| Activate document window | Esc | Window.ActivateDocumentWindow |
| Close document window | Ctrl+F4 | Window.CloseDocumentWindow |
| Next document window | Ctrl+F6 | Window.NextDocumentWindow |
| Next document window nav | Ctrl+Tab | Window.NextDocumentWindowNav |
| Next split pane | F6 | Window.NextSplitPane |