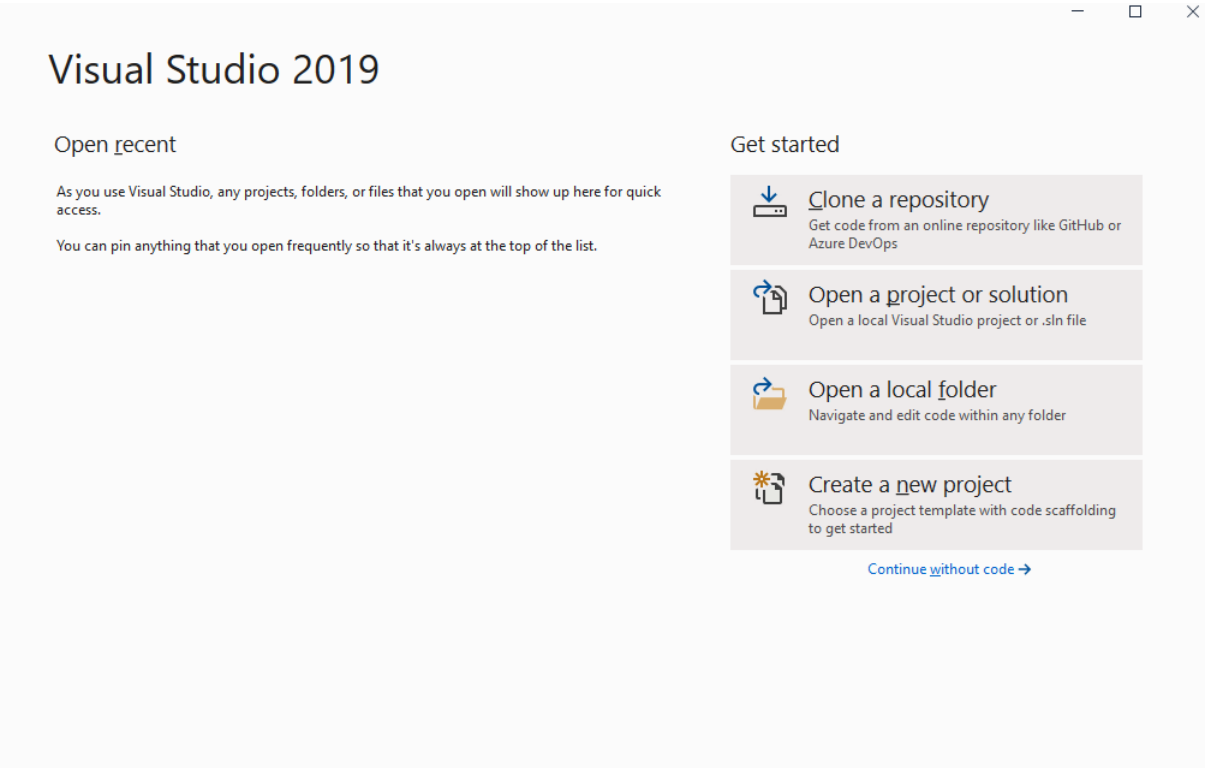
RKIT Module – 1

Name: Adhyaru Keyur D.

Visual Studio 2019 IDE Overview

Welcome Screen



1. **Clone a Repository**: We can clone the data from version control system.
    - Visual provide 2 types of Version control system.
        1. GitHub: (Microsoft) GitHub is free for public repository with 500 MB Packaging storage.
        2. Azure DevOps: Azure DevOps is free for open source projects and small Projects.

        It's has many advance features to work with.
        - Dashboard Control
        - Improved Source Control
        - Plan and track your work
2. **Open a project or Solution**: By click this we can open the project or solution file of local directories.
3. **Open a local folder**: We can open project folder from this menu.
4. **Create a new Project:** We can create a new project by clicking this menu.

Different types of Windows

1. **Solution Explorer:** Using Solution Explorer Window we can manage, create and update projects files and solution.
2. **Properties Window:** Displays all the properties of the Components.
3. **Team Explorer:** We can work with team in Visual Studio. Visual Studio Provides Two Version Control System. GitHub and Azure DevOps.
4. **Bookmark Window:** Displays all bookmarked lines of Code File Location and Line Number.
5. **Error list:** We can View the error and warnings here.
6. **Git Changes:** Here we can view all the Change done in the Source code.
7. **Toolbox:** Here we can find all the complements which is used to create application interface.
8. **Terminal:** we have executed the commend using this terminal window.

Solution and Project

Solution Explorer

- A Project is contained within a solution. Solution Explorer is a container for one or more related projects along with build information, Visual Studio window settings associated with a particular project.

Solution File

- Visual Studio uses two file types
    1. sln – Visual Studio Solution
    2. suo – Solution User Options

Projects

- When we Start Building Website or application in visual studio, first we need to create projects.
- Project Contains All files. which is Executable.
- Files includes source code, icons, images, data files etc.
- Visual Studio uses **MS Build** to build each project in a solution.
- The Microsoft Build Engine (MS Build) is a platform for building application.
- It provides XML schema for a project file.
- Types of Projects – C# Project (.csproj), Visual Basic Project (.vbproj), or a database project (.dbproj).

Code editor features

Git (Version Control):

- Using Version Control System, it's is easy to work with Team Projects.
- Visual Studio Provides Inbuilt Version control system so it's is easy to Work with. No Need to type commands.

Line Number:

- Now When we have built the large project and one single file has a thousand line of code then it is good to have a line number.
- Visual Studio also provide to move at particular line number.

Format Document:

- It's is Good Practical's to have a proper structure code.
- When we have a clean written code than it's is good to work with.
- Visual provide a feature to format our code in a proper style.

Syntax Colouring:

- Visual Studio colour the code in a proper format. All syntax and keywords are coloured in blur So it is good to identify the code structure.
- Bracket Matching:
- In Visual Studio, All the Brackets are matches from open to close.
- So it's is Good to identify the Brackets of Code.

Shortcuts

Key	Use
Format Document	Ctrl + K
New Project	Ctrl + Shift + N
Open Project	Ctrl + Shift + O
New File	Ctrl + N
Open File	Ctrl + O
Save As	Ctrl + Shift + S
View in Browser	Ctrl + Shift + W
Complete Word (get Suggestion)	Ctrl + Spacebar
Find word	Ctrl + F
Find and Replace Word	Ctrl + H
Save the File	Ctrl + S
Cut line	Ctrl + X
Copy Line	Ctrl + C
Move Line	Alt + (UP/Down)
Scroll Line	Ctrl + (UP/Down)
Go To Line	Ctrl + G
Rename	F2
Undo	Ctrl + Z
Redo	Ctrl + Y
Duplicate	Ctrl + D
Next tab	Ctrl + Tab
Find in Files	Ctrl + Shift + F
Breakpoint	F9
Debug Start / Continue	F5
Debug Stop	Shift + F5

Project Types

Windows App, Class Library

Windows App:

- Windows Forms is a Graphical User Interface(GUI).
- It is installed in the windows platform using windows operating system.
- Which is used to create or develop easier interface for applications of desktop, tablets etc.
- Windows Forms Application can contain the different type of controls like labels, list boxes etc.
- It can run only on windows platform.

How to Create a Windows Forms Application in visual Studio?

- ✓ Step 1: Open Visual Studio Go to File and then click on New.
- ✓ Step 2: Click on the Project. Here we have to choose the language as C#.
- ✓ Step 3: Now Select on the Windows Forms App and enter the project name and location to save.
- ✓ Step 4: Click on the Ok Button.

Class Library:

- Class library defines types and the methods that are called by an Applications.
- When we create a class library it can be distribute on NuGet Package or as a component bundled with the application that uses it.

How to Create Class Library Project?

- ✓ Step: 1 Create one project.
- ✓ Step 2: Right-Click on the Solution in Solution Explorer and select Add > New Project.
- ✓ Step 3: Choose C# language and click on the Class library.
- ✓ Step 4: Write the Project name and location of the project.
- ✓ Step 5: Click on the next button.

Web Application:

- Web Application is an application that runs on web browser making use of web server.
- Web application is a Client-side and server-side software application.
- We can run an application in browser.
- It can be accessed from anywhere around world using internet.
- Web application is independent of type of system.

How to Create Web Application?

- ✓ Step 1: Create a new Project.
- ✓ Step 2: Choose C# Language and choose windows in platform list.
- ✓ Step 3: Choose ASP.Net Core Web App form list.
- ✓ Step 4: Now Enter the Name and location of the project.
- ✓ Step 5: Click on the Next.

Create First C# Program "Hello World"

What is Namespace?

- namespaces are used to logically arrange classes, structs, interfaces, enums and delegates.
- The namespaces can be nested. That means one namespace can contain other namespaces also.
- The .NET framework already contains number of standard namespaces like System, System.Net, System.IO etc.
- In addition to these standard namespaces the user can define their own namespaces.

Declaring a Namespace

Syntax:

```
namespace <namespace_name>
{
        // Classes and/or structs and/or enums etc.
}
```

- **Note: namespace** keyword is used to define namespace
- It is not possible to use any access specifiers like private, public etc with a namespace declaration.
- The namespaces in C# are implicitly have public access and this is not modifiable.
- Default it provide internal access.

Creating Aliases

- Developers can create Aliases of the namespace.

Example:

```
using con = System.Console; // Create an alias
class MyClient
{
        public static void Main()
        {
                con.WriteLine("namespace demo");
        }
}
```

Standard Namespaces in .NET

- System: Contain classes that implement basic functionalities like mathematical operations, data conversions etc.
- System.IO: Contains classes used for file I/O operations.
- System.Net: Contains class wrappers around underlying network protocols.
- System.Collections: Contains classes that implement collections of objects such as lists, hashtable etc.
- System.Data: Contains classes that make up ADO.NET data access architecture.
- System,Drawing: Contains classes that implement GUI functionalities.
- System.Threading: Contains classes that are used for multithreading programming.
- System.Web: Classes that implement HTTP protocol to access web pages.
- System.Xml: Classes that are used for processing XML data.

We can use namespace using two methods.

1. Dot(.) Operator
2. Using directive

What is class?

- A class is like a **blueprint** of a specific object.
- In real world every object has some colour, shape and some functionalities etc.
- It has also some characteristics like speed, colour etc.
- So Group of some this all things is called class.
- A class enables you to create your custom types by grouping variables of the other types, methods and events.
- Class can be defined in **class** keyword.

Access Modifier

Public: Public modifier allows any part of the program in the same assembly or another assembly to access the type and its members.

Private: Private modifier restricts other parts of the program from accessing the type and its members. Only code in the same class or struct can access it.

Internal: Internal modifier allows other program code in the same assembly to access the type or its members. This is default access modifiers if no modifier is specified.

Protected: Protected modifier allows codes in the same class or a class that derives from that class to access the type or its members.

Constructor

- Class have parameterized or parameter less constructors.
- Constructor will be called when you create an instance of a class.
- Constructor will be defined by using as access modifier.

Syntax:

```
class MyClass
{
   public MyClass()
   {

   }
}
```

Variable & Method Declaration

Variables:

- variable contains a data value of the specific data type

Syntax:

```
<data type> <variable name> = <value>;
```

Example:

```
int age = 10;
```

- There are some rules to declare variables name.
- Variable names must be unique.
- Variable names can contain letters, digits, and the underscore _ only.
- Variable names must start with a letter.
- Variable names are case-sensitive, num and Num are considered different names.
- Variable names cannot contain reserved keywords. Must prefix @ before keyword if want reserve keywords as identifiers.

Method

- A program causes the statements to be executed by calling the method.
- Method is use redenied code.
- A method can also perform some specific task without returning anything.

Syntax:

{access modifier} {return type} MethodName({parameterType parameterName})

- Method have some elements explain below
- Return Type: We have to specify the return type of the method.
- Method Body: This is the actual part of code which is execute when method called.
- Parameters Body: We have to specify the method parameters.
- Access Specifier: set the visibility of a variable or a method from another class.

Understanding C# Program

Program Flow

- Generally, Programs Contains following Parts

Namespace declaration
Class
Class Methods
Class Attributes
Statements and Expressions
Comment

- Now in C# Programs is Executing Task in Sequence manner
- Comments are not considered at the time of application.

Understanding Syntax

- C# follow the standard structure for the code.
- C# is a Cass-Sensitive Language.
- It Follow the Below Structure.

Imports / namespace	Using keyword is use to declare namespace.
Declare namespace	Namespace declare for the class
Class declare	Class declare with proper name
Methods of class	Other methods of class
Main() Method	Main method is the method by which program execute.

Working with code files, projects & solutions

Understanding structure of solution

Solutions:

- A Solution contains a collection of projects, along with information on dependencies between those projects.
- The project themselves contain files.
- We can Create many projects in solution
- We can one open one solution at a time in a particular instance of visual studio.

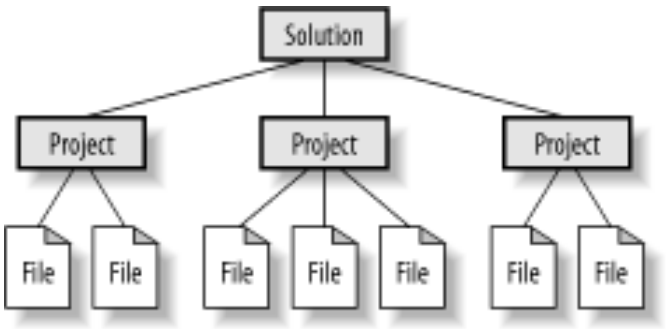


Figure show the Solution structure

- One project can have multiple solutions.

Understanding structure of Project

Windows App:

- Windows application is used widely.
- There are many architectural for windows application.

Controls:

- In the Windows application there are many controls like textbox, edit text menu etc.
- This are used to complete the functionally of users.

Forms: This is the actually pages of Windows Application

Dependencies:

- Here we can find the projects dependencies.

Program.cs:

- This is the Main Entry Point of the application.

Web Applications:

- Using Web application, we can build a website.
- In the Website Project there is many folders.

Appsettings.json:

- Appsettings json file is an application configuration file used to store configuration settings such as database connections strings.
- Using this file, we can declare the value globally.

Dependencies:

- Here we can find the projects dependencies.

Wwwroot folder:

- CSS: this folder has the css file of the website.
- JS: This folder has the JavaScript files of the website.

Project.json file:

- It contains the information about the project.

Familiar with different type of file extensions

Name	Use
.html,.htm	Html web file
.asp	Active server page file
.css	Cascading Style Sheets
.txt	A text file
.sln	Solution file
.csproj	C# Project
.png	A image file

Understanding datatypes & variables with conversion

Base Datatypes

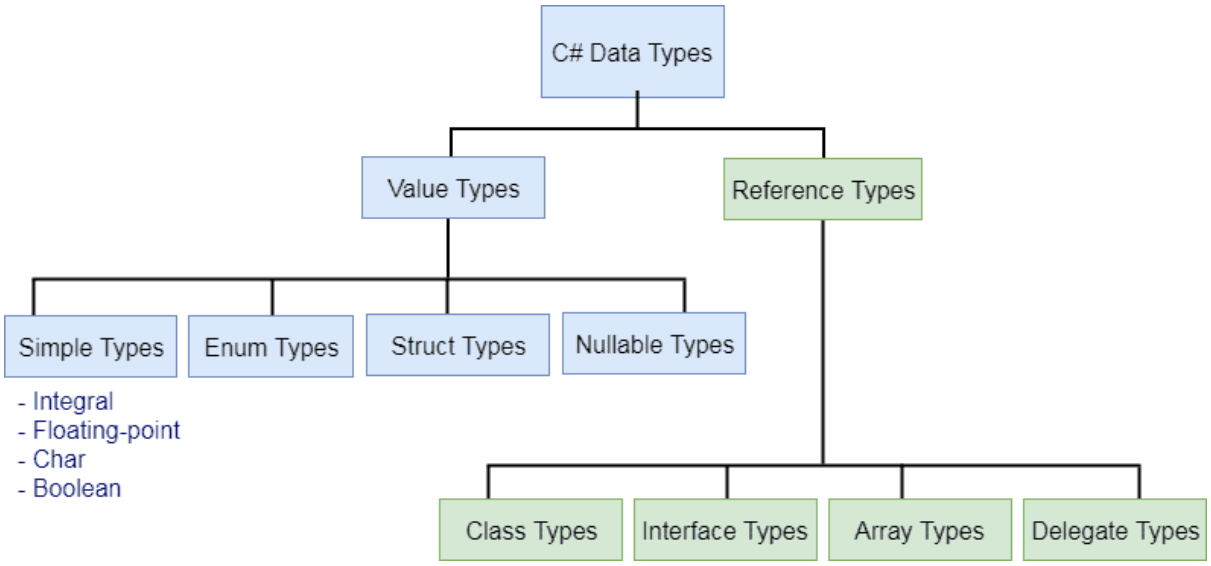
- C# is a strongly-typed language.
- It means we must declare the type of a variable that indicates that kind of values.
- Like int, float, text, etc.
- C# has two type of data types (value and reference type)

Value Datatype:

- Value datatype store the data itself.
- Value types include simple types such as int, float, bool.

Reference Type:

- Reference type doesn't store its vale directly.
- It stores the address where the value is being stored.
- Reference type contains a pointer to another memory location that holds the data.



Datatype Conversion

- C# has a two type of Conversion of datatype.
- It is also known as type Casting.
- Two types are Implicit Type Conversion and Explicit Type Conversion.

1. Implicit Type Conversion:

- These conversions are performed by C# in a type-safe manner.
- From the smaller to larger integral types and conversions from derived classes to base classes.

2. Explicit Type Conversion:

- These conversions are done explicitly by users using the predefined functions.
- Explicit Conversion require a cast operator.

Methods	Description
ToBoolean	Converts a type to a Boolean value, where possible.
ToByte	Converts a type to a byte
ToChar	Converts a type to a single Unicode character, where possible
ToDateTime	Converts a type (integer or string type) to date-time structures.
ToDecimal	Converts a floating point or integer type to a decimal type.
ToDouble	Converts a type to a double type
ToInt16	Converts a type to a 16-bit integer.
ToInt32	Converts a type to a 32-bit integer.
ToInt64	Converts a type to a 64-bit integer.
ToString	Converts a type to a string.

Boxing and Unboxing

- In C#, there are two Kinds of datatype.
  - ✓ Value type (int, double, decimal).
  - ✓ Reference type (object, string, array).
- **Value Type:** value type stores the value itself.
- **Reference Type:** reference type stores the address of the value where it is stored.

What is Boxing?

- Process of converting a value type to the object type.
- Boxing is implicit.

Example:

```
int i = 10;

object o = i ;
```

- In the Above Example we have did Boxing. Now Object O store the address of the variable i.
- So This is called as Boxing.
- Now its Name is Boxing because it's store the address in heap and actual value is store in the stack So heap wrap the value and box on this value and address is store in the heap so It's is Called Boxing.

What is Unboxing?

- Unboxing is the reverse of boxing.
- It is the process of converting a reference type to value type.
- Unboxing is Explicit. It's Means we need to cast explicitly.

Example:

```
Int I = 10

Object o = i;

Console.WriteLine((int)0);
```

- **Note: Boxing and Unboxing degrade the performance.**

If-else & Switch Statement

If-Else:

- We can get Many decision-making statements in C#
- like if, else if and else
- below is the syntax.

If Statement

```
if(condition)
{
    // code block to be executed when if condition evaluates to true
}
```

If Else Statement

```
if(condition1)
{
    // code block to be executed when if condition1 evaluates to true
}
else if(condition2)
{
    // code block to be executed when
    //      condition1 evaluates to flase
    //      condition2 evaluates to true
}
```

Example:

```
int i = 10;


if (i > 3)
{
   Console.WriteLine("i is greater than 3");
}

if (i >= 10)
{
   Console.WriteLine("i is greater than or equal to 10");
}
```

Switch Statement:

- Switch Statements Start with **switch** Keyword.
- Switch Statements is use to execute possible block of Code.
- Switch expression is evaluated once.
- It has one Default case also
- Default keyword is optional and specifies some code to run if there is no case match.

Syntax:

```
switch(match expression/variable)
{
    case constant-value:
        statement(s) to be executed;
        break;
    default:
        statement(s) to be executed;
        break;
}
```