# DAY 2 : Latest C# and .NET 5 Features

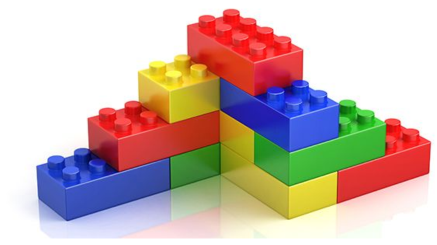
Tuesday, October 12, 2021 10:07 AM

| |
|---|
| • Creating Overloaded Methods and Using Optional and Output Parameters (Positional /Optional parameters later will be covered) |
| • Handling Exceptions |
| • Monitoring Applications |
| • Implementing Structs and Enums |
| • Organizing Data into Collections |
| • Handling Events |
| • Creating Classes |
| • Defining and Implementing Interfaces (More Ex Will be covered later) |
| • Implementing Type-Safe Collections , <Templates> Generic Collection |

NGWS : Next generation Windows Services
.NET (NETWORK ENABLED TECHNOLOGIES)
.net - network (Wrong)

JAVA/HTML5:

In java we have Lego Blocks :



.NET came thought process

.NET Version = 5

-What All are latest Addition to it :

- .NET 5.0 is the next major release of .NET Core following 3.1. We named this new release .NET 5.0 instead of .NET Core 4.0 for two reasons:
- We skipped version numbers 4.x to avoid confusion with .NET Framework 4.x.
- We dropped "Core" from the name to emphasize that this is the main implementation of .NET going forward. .NET 5.0 supports more types of apps and more platforms than .NET Core or .NET Framework.
- ASP.NET Core 5.0 is based on .NET 5.0 but retains the name "Core" to avoid confusing it with ASP.NET MVC 5. Likewise, Entity Framework Core 5.0 retains the name "Core" to avoid confusing it with Entity Framework 5 and 6.

Lang Version C# version :

- Simple, Latest, Flexible and Modern Programming lang.
- Genral Purpose Object oriented lang

-What New You can do with Latest Lang Features :

- Records
- Init only setters
- Top-level statements
- Pattern matching enhancements
- Performance and interop
  - Native sized integers
  - Function pointers
  - Suppress emitting localsinit flag
- Fit and finish features
  - Target-typed new expressions
  - static anonymous functions
  - Target-typed conditional expressions
  - Covariant return types
  - Extension GetEnumerator support for foreach loops
  - Lambda discard parameters
  - Attributes on local functions
- Support for code generators
  - Module initializers
  - New features for partial methods

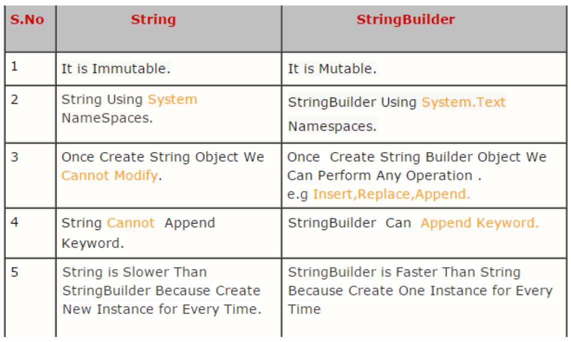
CLR version =

ASP.NET V5 | ASP.NET Core Version 5
ASP.NET MVC | ASP.NET Core MVC

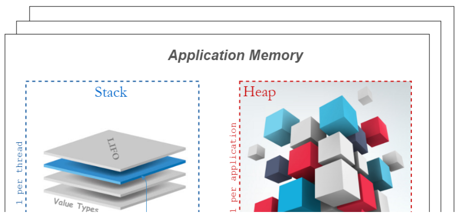
public record Person( string Fname, String LName);

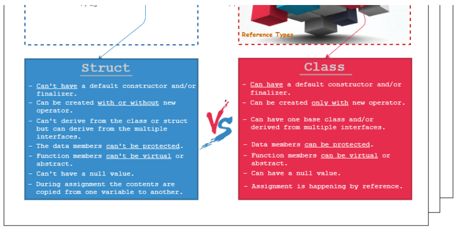
String :
string : This is same

But IN terms –f String and String Builders following are the differences



| S.No | String | StringBuilder |
|---|---|---|
| 1 | It is Immutable. | It is Mutable. |
| 2 | String Using System NameSpaces. | StringBuilder Using System.Text Namespaces. |
| 3 | Once Create String Object We Cannot Modify. | Once Create String Builder Object We Can Perform Any Operation . e.g Insert,Replace,Append. |
| 4 | String Cannot Append Keyword. | StringBuilder Can Append Keyword. |
| 5 | String is Slower Than StringBuilder Because Create New Instance for Every Time. | StringBuilder is Faster Than String Because Create One Instance for Every Time |





**Struct**

- Can't have a default constructor and/or finalizer.
- Can be created with or without new operator.
- Can't derive from the class or struct but can derive from the multiple interfaces.
- The data members can't be protected.
- Function members can't be virtual or abstract.
- Can't have a null value.
- During assignment the contents are copied from one variable to another.

**Class**

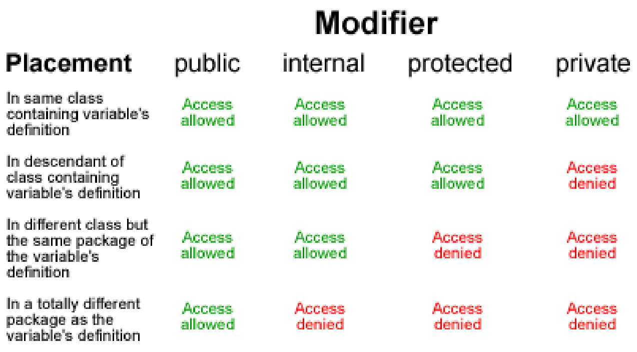
- Can have a default constructor and/or finalizer.
- Can be created only with new operator.
- Can have one base class and/or derived from multiple interfaces.
- Data members can be protected.
- Function members can be virtual or abstract.
- Can have a null value.
- Assignment is happening by reference.

| **Structure** | **Class** |
|---|---|
| It is a value type. | It is a reference type. |
| Its object is created on the stack memory. | Its object is created on the heap memory. |
| It does not support inheritance. | It supports inheritance. |
| The member variable of structure cannot be initialized directly. | The member variable of class can be initialized directly. |
| It can have only parameterized constructor. | It can have all the types of constructor and destructor. |

**When To use Structure over class**

1. When we are dealing with Value Types
2. When we want faster Access and abundance of memory.
3. When we don't want inheritance to take place
4. When we want to provide code security with Read only properties.

We have Following  Modifiers in C#



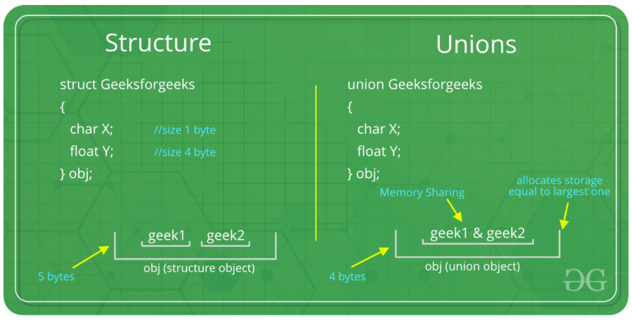
| | **Modifier** | | | |
|---|---|---|---|---|
| **Placement** | public | internal | protected | private |
| In same class containing variable's definition | Access allowed | Access allowed | Access allowed | Access allowed |
| In descendant of class containing variable's definition | Access allowed | Access allowed | Access allowed | Access denied |
| In different class but the same package of the variable's definition | Access allowed | Access allowed | Access denied | Access denied |
| In a totally different package as the variable's definition | Access allowed | Access denied | Access denied | Access denied |

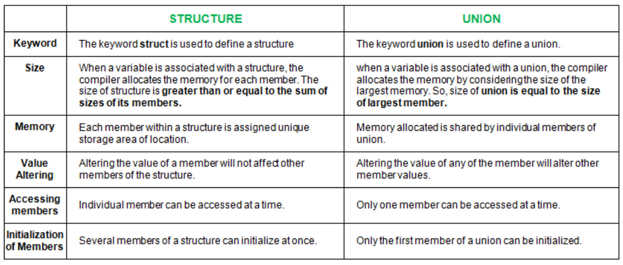
Why we need Enum in C# ? - reference Types

- When we want to implement predefine variables.
- Where value is Not frequently changing
- All Flexibity of a typical reference type user define variables.
- Ex Group of Employes with Serial No can Implemed as enum
- It supports all Access Modifiers

**System.enum** class which is abstract base class.

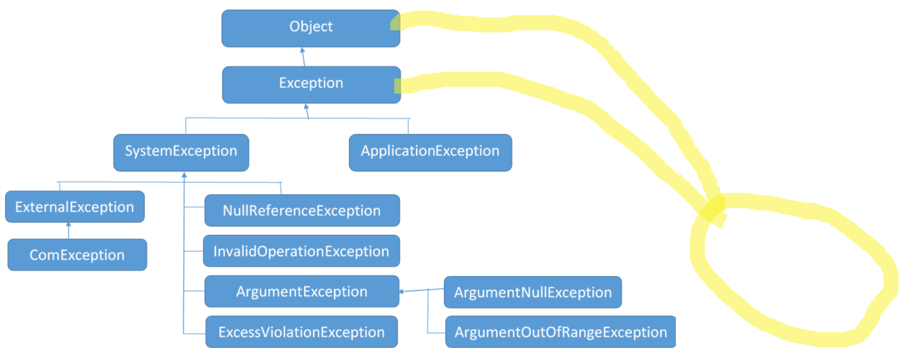
A boxing conversion (Boxing conversions) exists from any enum type to `System.Enum`, and an unboxing conversion (Unboxing conversions) exists from `System.Enum` to any enum type.

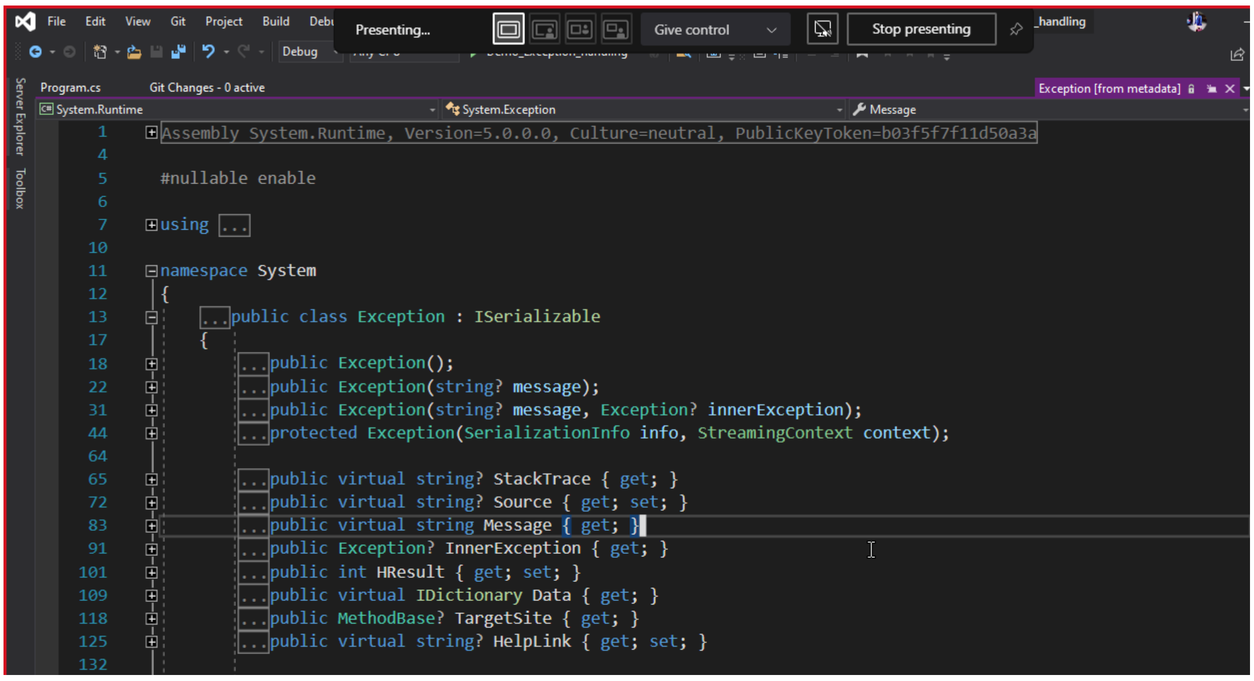




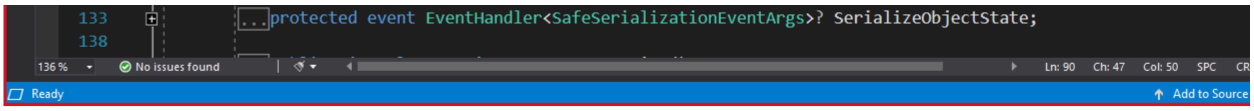
| | STRUCTURE | UNION |
|---|---|---|
| **Keyword** | The keyword **struct** is used to define a structure | The keyword **union** is used to define a union. |
| **Size** | When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is **greater than or equal to the sum of sizes of its members.** | when a variable is associated with a union, the compiler allocates the memory by consideringthe size of the largest memory. So, size of **union is equal to the size of largest member.** |
| **Memory** | Each member within a structure is assigned unique storage area of location. | Memory allocated is shared by individual members of union. |
| **Value Altering** | Altering the value of a member will not affect other members of the structure. | Altering the value of any of the member will alter other member values. |
| **Accessing members** | Individual member can be accessed at a time. | Only one member can be accessed at a time. |
| **Initialization of Members** | Several members of a structure can initialize at once. | Only the first member of a union can be initialized. |

Exception handling in C#
Try ....Catch... Finally...Throw...





```
[Assembly System.Runtime, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a

#nullable enable

using ...

namespace System
{
    public class Exception : ISerializable
    {
        public Exception();
        public Exception(string? message);
        public Exception(string? message, Exception? innerException);
        protected Exception(SerializationInfo info, StreamingContext context);

        public virtual string? StackTrace { get; }
        public virtual string? Source { get; set; }
        public virtual string Message { get; }
        public Exception? InnerException { get; }
        public int HResult { get; set; }
        public virtual IDictionary Data { get; }
        public MethodBase? TargetSite { get; }
        public virtual string? HelpLink { get; set; }
```



```
protected event EventHandler<SafeSerializationEventArgs>? SerializeObjectState;
```

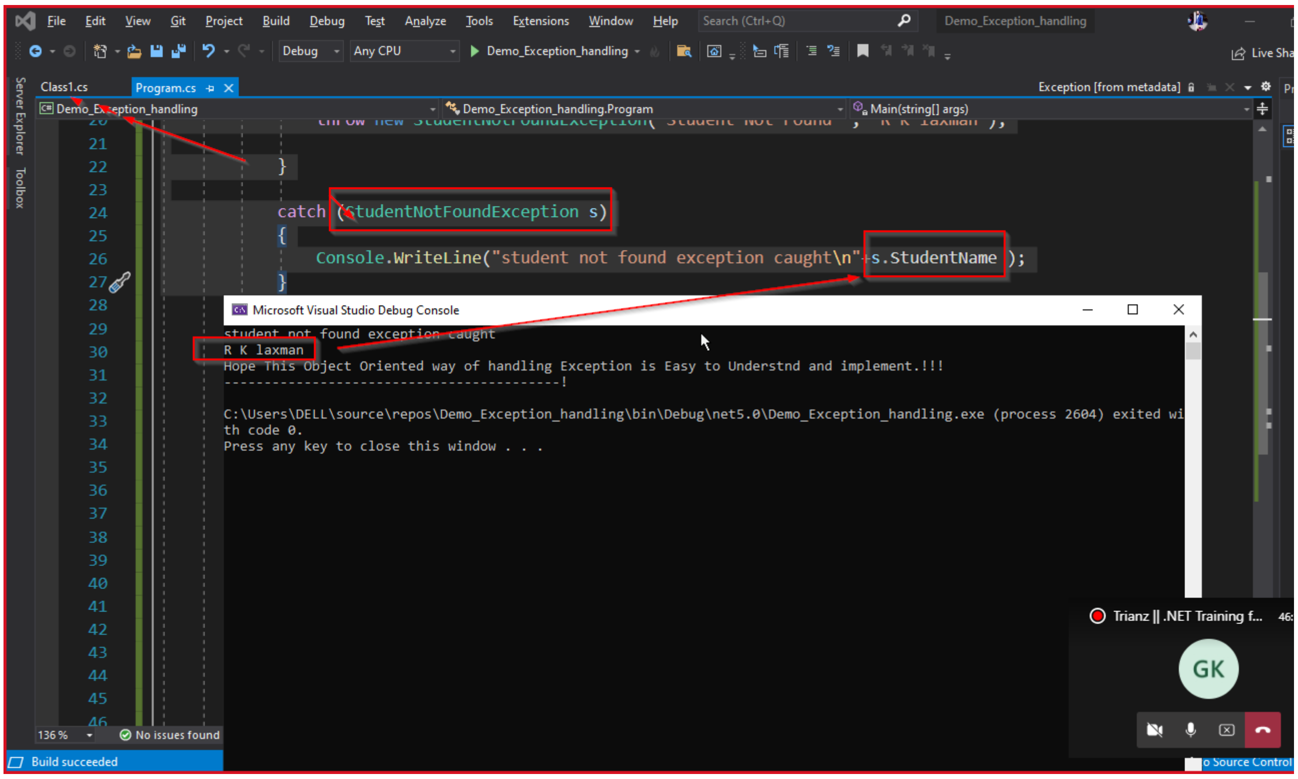
Writing CUSTOM DEFINE EXCEPTION USING
EXCEPTION BASE CLASS

```
[Serializable]
    public class StudentNotFoundException :
Exception
    {
        public string StudentName { get; set; }
        public StudentNotFoundException() { }
        public StudentNotFoundException(string
message) : base(message) {}

        //Console.WriteLine(" Calling base class
message defined in Exception class");

        public StudentNotFoundException(string
Message, Exception InnerException) :
base(Message, InnerException) { }

        public StudentNotFoundException(string
message, string studentName)
        : this(message)
        {
            StudentName = studentName;
        }
```

ON main() File

```
try
            {
                //Console.WriteLine("Enter your Favourate
No");
                //string No =  Console.ReadLine();
                //Int32.Parse(No);

                //Console.WriteLine("No You entered {0}.
valid Integer 32", No);

                throw new
StudentNotFoundException("Student Not Found ",
"R K laxman");

            }
```

```
        catch (StudentNotFoundException s)
        {
            Console.WriteLine("student not found
exception caught\n"+s.StudentName );
        }
```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Demo_Exception_handling
Debug
Any CPU
Class1.cs
Program.cs
Exception [from metadata]
Demo_Exception_handling.Program
Main(string[] args)
Microsoft Visual Studio Debug Console
Trianz || .NET Training f...
GK
136 %
No issues found
Build succeeded
Source Control

```
        }

        catch (StudentNotFoundException s)
        {
            Console.WriteLine("student not found exception caught\n"+s.StudentName );
        }
```

```
student not found exception caught
R K laxman
Hope This Object Oriented way of handling Exception is Easy to Understnd and implement.!!!
----------------------------------------!

C:\Users\DELL\source\repos\Demo_Exception_handling\bin\Debug\net5.0\Demo_Exception_handling.exe (process 2604) exited wi
th code 0.
Press any key to close this window . . .
```

Base() this refers to definition of variables defined in the base Class

Overrides() this overrides definition of Base Class Function by Child Class

**<u>How to use User define Exception in Your Project :</u>**

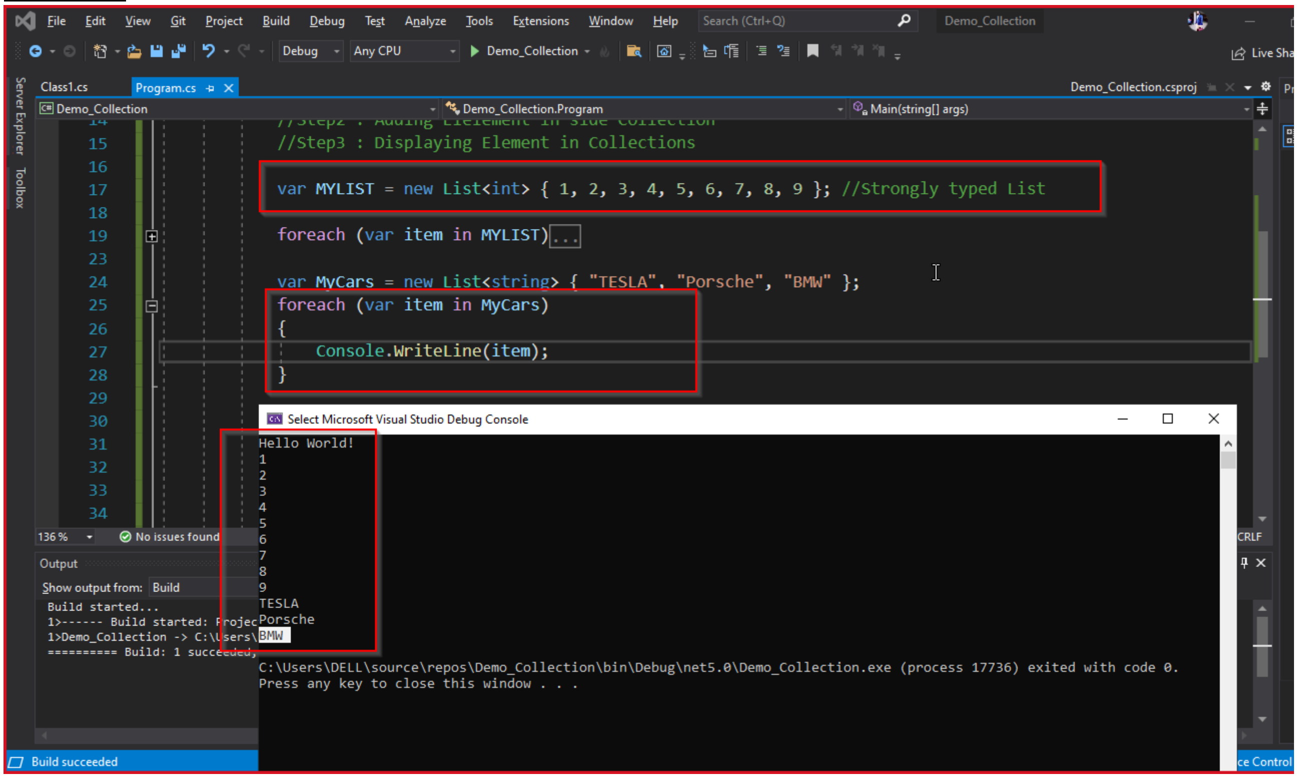
- Game Development Where all Boundary Conditions(Game over) are dependent on Problem Statement
  - When head pf the snake Touches its body
  - When head of the Snake Touches Boundry (Frame)
- When we want to avoid Sudden halt of Code Execution

When We have Huge Predefine and user define Datatype why we should use C# Collections ?

1. We can have Same name pointing to Multiple Memory locations.

2. Predefine Methods are there in collections classes.
3. Specific set of templating helps us in storing particular set of value.

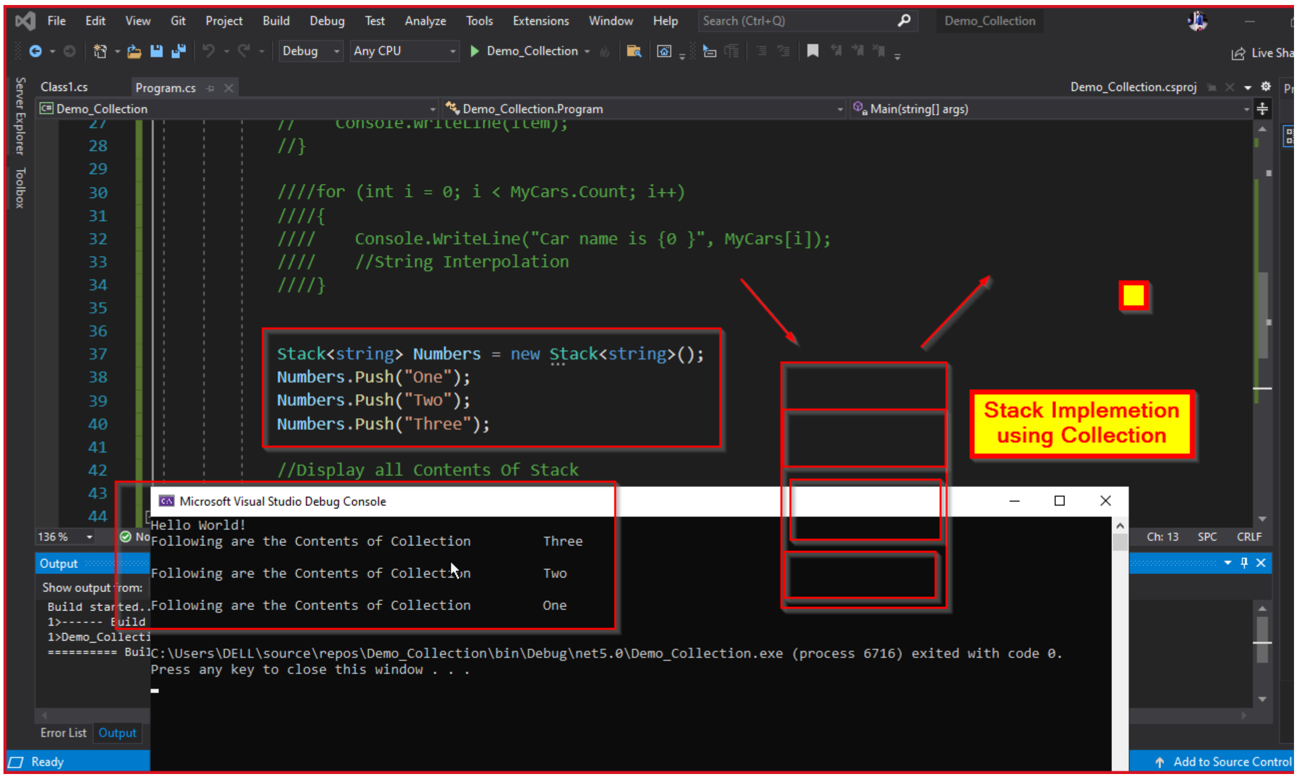
## Working with Array List<Template> generic Collection



```
//Step2 : Adding Element in side Collection
//Step3 : Displaying Element in Collections

var MYLIST = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9 }; //Strongly typed List

foreach (var item in MYLIST)...

var MyCars = new List<string> { "TESLA", "Porsche", "BMW" };
foreach (var item in MyCars)
{
    Console.WriteLine(item);
}
```

```
Build started...
1>------ Build started: Projec
1>Demo_Collection -> C:\Users\
========== Build: 1 succeeded,
```

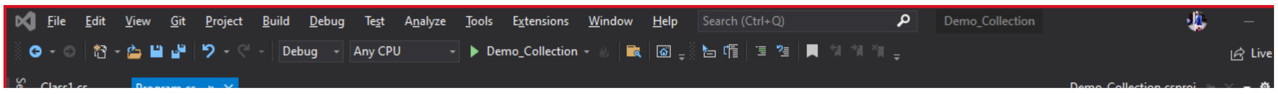
```
Hello World!
1
2
3
4
5
6
7
8
9
TESLA
Porsche
BMW

C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 17736) exited with code 0.
Press any key to close this window . . .
```

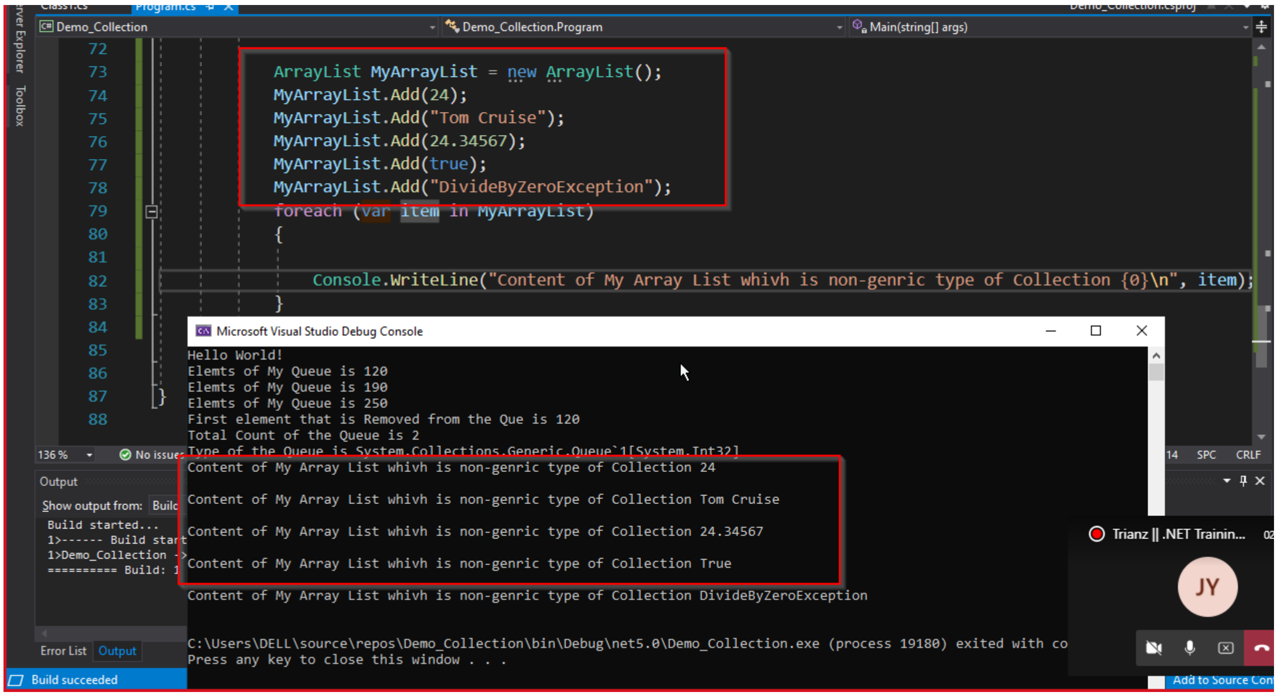


```
//    Console.WriteLine(item);
//}

////for (int i = 0; i < MyCars.Count; i++)
////{
////    Console.WriteLine("Car name is {0 }", MyCars[i]);
////    //String Interpolation
////}

Stack<string> Numbers = new Stack<string>();
Numbers.Push("One");
Numbers.Push("Two");
Numbers.Push("Three");

//Display all Contents Of Stack
```

```
Hello World!
Following are the Contents of Collection        Three

Following are the Contents of Collection        Two

Following are the Contents of Collection        One

C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 6716) exited with code 0.
Press any key to close this window . . .
```

When to use Collections in C# ?

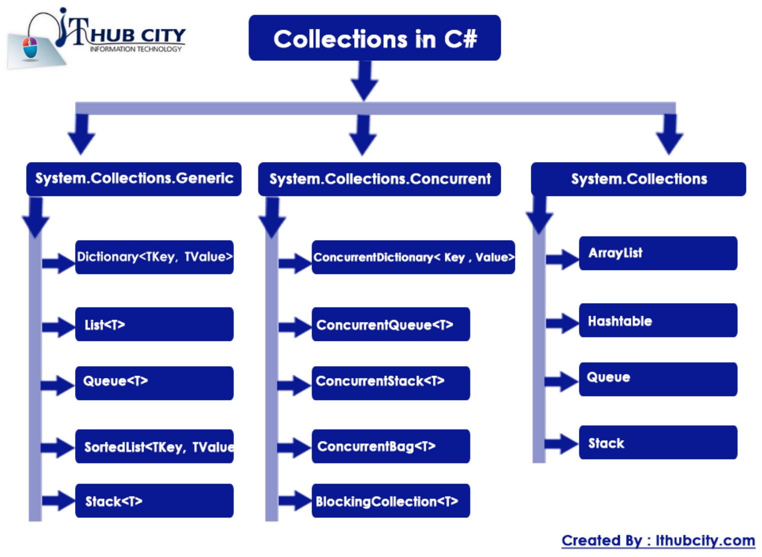




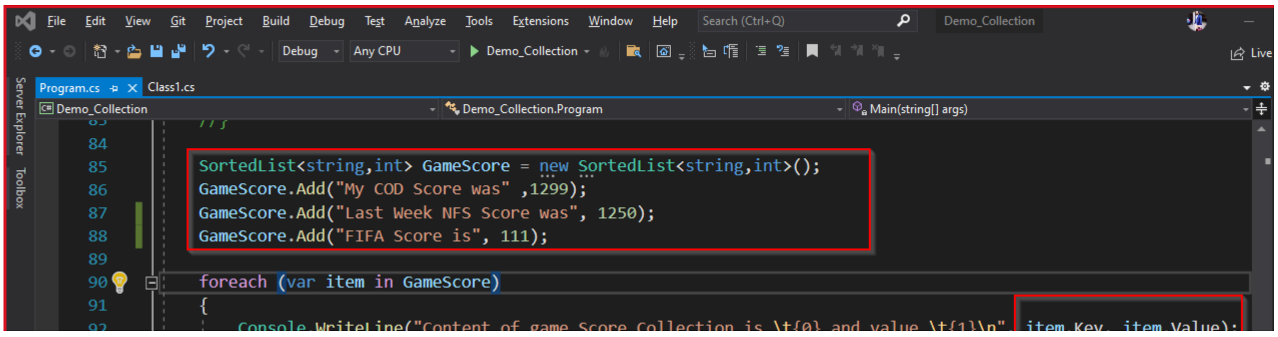
```
ArrayList MyArrayList = new ArrayList();
MyArrayList.Add(24);
MyArrayList.Add("Tom Cruise");
MyArrayList.Add(24.34567);
MyArrayList.Add(true);
MyArrayList.Add("DivideByZeroException");
foreach (var item in MyArrayList)
{
    Console.WriteLine("Content of My Array List whivh is non-genric type of Collection {0}\n", item);
}
```

```
Hello World!
Elemts of My Queue is 120
Elemts of My Queue is 190
Elemts of My Queue is 250
First element that is Removed from the Que is 120
Total Count of the Queue is 2
Type of the Queue is System.Collections.Generic.Queue`1[System.Int32]
Content of My Array List whivh is non-genric type of Collection 24

Content of My Array List whivh is non-genric type of Collection Tom Cruise

Content of My Array List whivh is non-genric type of Collection 24.34567

Content of My Array List whivh is non-genric type of Collection True

Content of My Array List whivh is non-genric type of Collection DivideByZeroException


C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 19180) exited with co
Press any key to close this window . . .
```

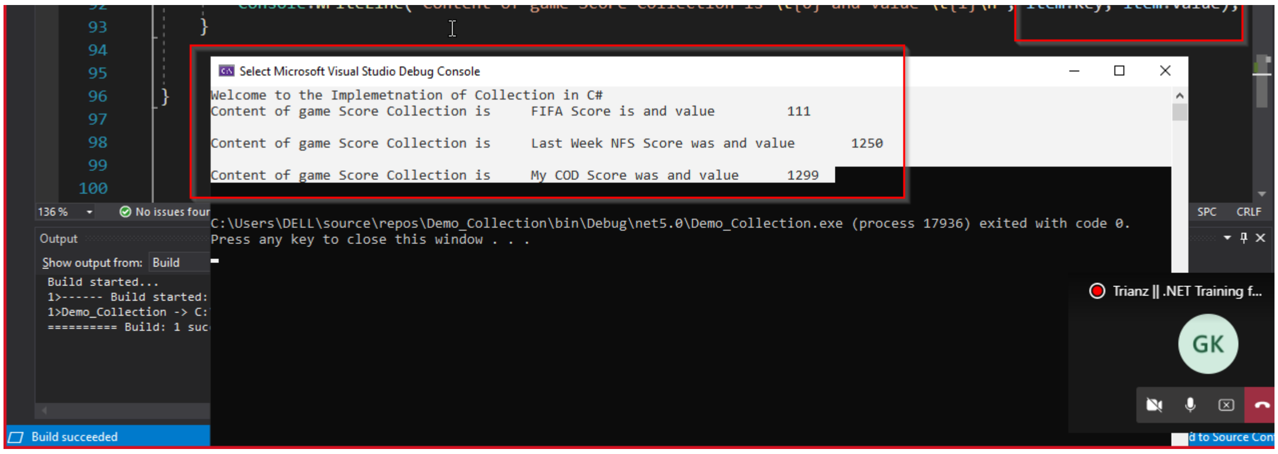
Steps for implementing on Generic Collection :
Step1 : Implementing
System.collection (namespace)
Step2 : Creating Object of the Class
Step3 : Adding element of Different type (Object)



Implementing Dictionary In C#



```
SortedList<string,int> GameScore = new SortedList<string,int>();
GameScore.Add("My COD Score was" ,1299);
GameScore.Add("Last Week NFS Score was", 1250);
GameScore.Add("FIFA Score is", 111);

foreach (var item in GameScore)
{
    Console.WriteLine("Content of game Score Collection is \t{0} and value \t{1}\n", item.Key, item.Value);
```



```
Welcome to the Implemetnation of Collection in C#
Content of game Score Collection is     FIFA Score is and value         111

Content of game Score Collection is     Last Week NFS Score was and value       1250

Content of game Score Collection is     My COD Score was and value      1299

C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 17936) exited with code 0.
Press any key to close this window . . .
```

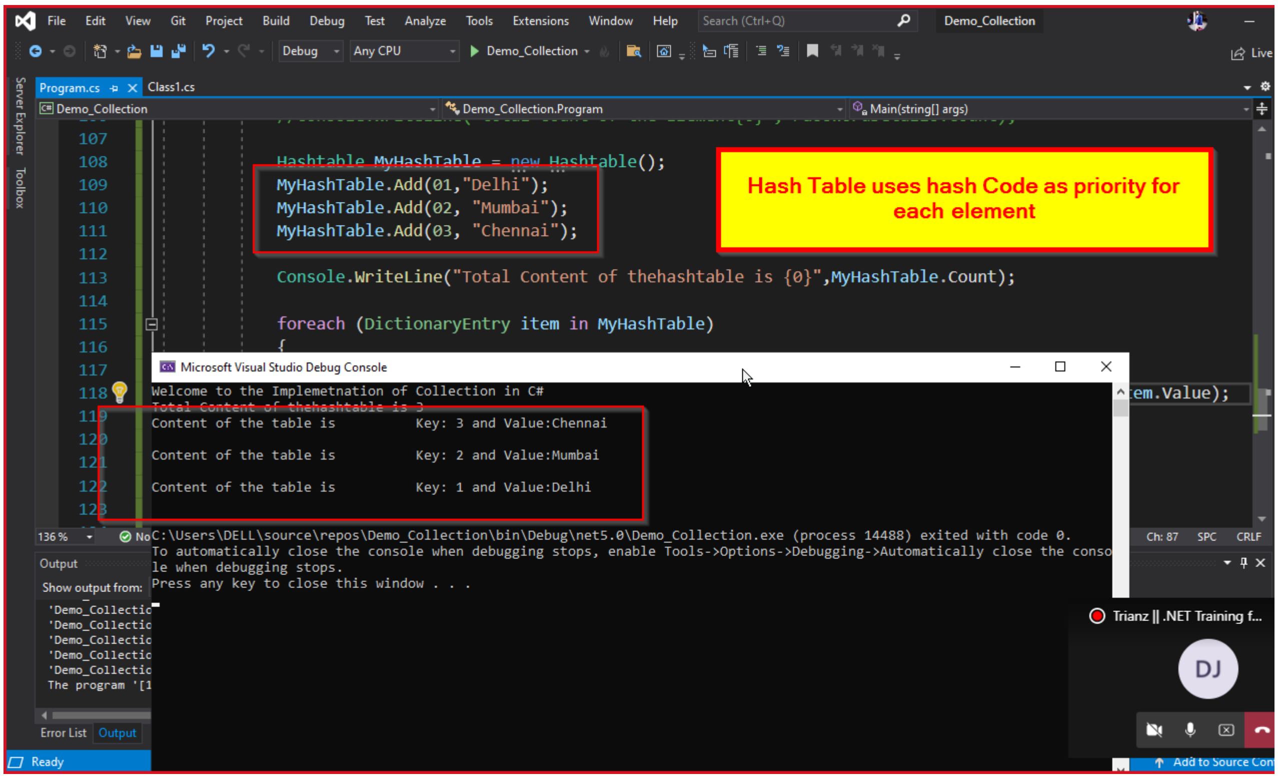
Select Microsoft Visual Studio Debug Console
Output
Show output from: Build
Build started...
Build succeeded
Trianz || .NET Training f...
GK

Real World Ex Of Dictionary :

- Storing Lib Books <name of the Books ,Author>
- <Expenses Amount, Date of transaction>
- <Process iD , Bug>
- <Process name , Duration>
- <DateTime Stamp, Transaction ID>

Technical Ex:

- <Username, password>
- <Name, AcessType>

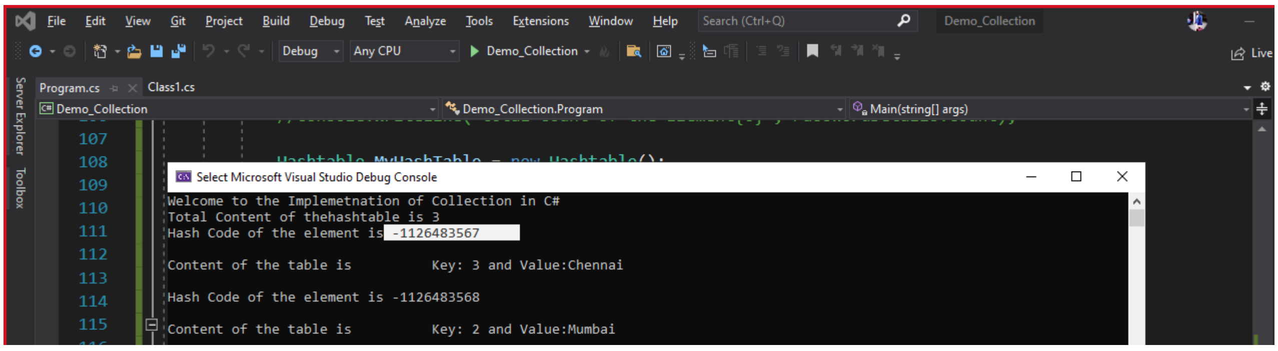


```csharp
Hashtable MyHashTable = new Hashtable();
MyHashTable.Add(01,"Delhi");
MyHashTable.Add(02, "Mumbai");
MyHashTable.Add(03, "Chennai");

Console.WriteLine("Total Content of thehashtable is {0}",MyHashTable.Count);

foreach (DictionaryEntry item in MyHashTable)
{
```

Hash Table uses hash Code as priority for each element

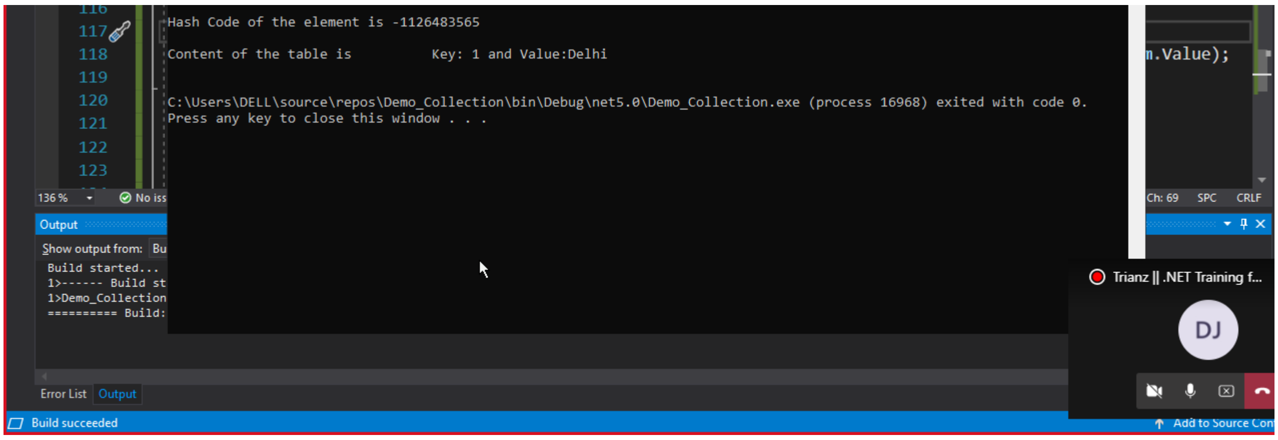
```
Welcome to the Implemetnation of Collection in C#
Total Content of thehashtable is 3
Content of the table is          Key: 3 and Value:Chennai

Content of the table is          Key: 2 and Value:Mumbai

Content of the table is          Key: 1 and Value:Delhi

C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 14488) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Demo_Collection
Debug
Any CPU
Program.cs
Class1.cs
Demo_Collection.Program
Main(string[] args)
Microsoft Visual Studio Debug Console
Output
Error List
Ready
Trianz || .NET Training f...
DJ



```
Welcome to the Implemetnation of Collection in C#
Total Content of thehashtable is 3
Hash Code of the element is -1126483567

Content of the table is          Key: 3 and Value:Chennai

Hash Code of the element is -1126483568

Content of the table is          Key: 2 and Value:Mumbai
```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Demo_Collection
Program.cs
Class1.cs
Demo_Collection.Program
Main(string[] args)
Select Microsoft Visual Studio Debug Console



```
Hash Code of the element is -1126483565

Content of the table is          Key: 1 and Value:Delhi


C:\Users\DELL\source\repos\Demo_Collection\bin\Debug\net5.0\Demo_Collection.exe (process 16968) exited with code 0.
Press any key to close this window . . .
```

```
Build started...
1>------ Build st
1>Demo_Collection
========== Build:
```

When We Want to Have a Unique identifier attached with a Value : We use hash Code.

Art of Adding Functionality to a Project With a View of Adding Layers : Interface

Interface members are public by default, and you can explicitly specify accessibility modifiers, such as `public`, `protected`, `internal`, `private`, `protected internal`, or `private protected`

**Every layer Will help us in Retaining and Implementing new Functionality**

**Task for the day**
**LMS - Learning management system**
**We Will Be Listing Out all the**

1. **Classes**
   a. **Functions()**
2. **Interfaces**
   a. **Functions()**
3. **Types of Collection that can be Implemented**
4. **Abstract/Static/partial Classes**

**Two main Modules in the System**

- **Assessment for the employee**
- **Final Report after assessment**