<https://www.c-sharpcorner.com/UploadFile/84c85b/using-generics-with-C-Sharp/>

<https://www.tutorialsteacher.com/csharp/csharp-generics>

<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/generics/generic-interfaces>

<https://www.c-sharpcorner.com/article/c-sharp-list/#:~:text=List%20class%20in%20C%23,remove%20items%20to%20a%20list>.

<https://www.bytehide.com/blog/icollection-csharp>

using System;

using System.Collections.Generic;

namespace GenericApp

{

public class TestClass<T>

{

// define an Array of Generic type with length 5

T[] obj = new T[5];

int count = 0;

// adding items mechanism into generic type

public void Add(T item)

{

//checking length

if (count + 1 < 6)

{

obj[count] = item;

}

count++;

}

//indexer for foreach statement iteration

public T this[int index]

{

get { return obj[index]; }

set { obj[index] = value; }

}

}

class Program

{

static void Main(string[] args)

{

//instantiate generic with Integer

TestClass<int> intObj = new TestClass<int>();

//adding integer values into collection

intObj.Add(1);

intObj.Add(2);

intObj.Add(3); //No boxing

intObj.Add(4);

intObj.Add(5);

//displaying values

for (int i = 0; i < 5; i++)

{

Console.WriteLine(intObj[i]); //No unboxing

}

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

namespace GenericApp

{

class Program

{

//Generic method

static void Swap<T>(ref T a, ref T b)

{

T temp;

temp = a;

a = b;

b = temp;

}

static void Main(string[] args)

{

// Swap of two integers.

int a = 40, b = 60;

Console.WriteLine("Before swap: {0}, {1}", a, b);

Swap<int>(ref a, ref b);

Console.WriteLine("After swap: {0}, {1}", a, b);

string a1 = "aaa"

, b1 = "bbb";

Swap<string>(ref a1, ref b1);

Console.ReadLine();

}

}

}

using System;

using System.Text;

using System.Collections.Generic;

namespace GenericApp

{

public class emp

{

private string name;

private int salary;

public emp(string name, int salary)

{

this.name = name;

this.salary = salary;

}

public override string ToString()

{

StringBuilder sb = new StringBuilder(200);

sb.AppendFormat("{0},{1}", name, salary);

return sb.ToString();

}

}

public class Program

{

static void Main(string[] args)

{

//define Dictionary collection

Dictionary<string, emp> dObj = new Dictionary<string, emp>(2);

//add elements to Dictionary

emp tom = new emp("tom", 2000);

dObj.Add("tom", tom); // key,value

emp john = new emp("john", 4000);

dObj.Add("john", john);

//print data

foreach (Object str in dObj.Values)

{

Console.WriteLine(str);

}

Console.ReadKey();

}

}

}