**Session 10: Params And Indexer.**

**Params: -** its used as parameter. It take the number of argument. Don’t have the knowledge about no of parameter. Only one params keyword is used in function declaration. Params keyword is used. Params length of zero if no argument. It work as array of object. It change the element in no of array.

**Indexer:** indexer is allow an object to be indexed such as an array. U can use [ ] to access indexer. Its similar to **property**.Properties required name but indexer doesn’t. access modifiers are used with this. This keyword is used to access the insatace. Indexers can be overloaded. Indexer are special types of property. Indexer has return type and access modifier. Regular member of class. Overloadin of indexer will possible.

S10\_\_ParamsAndIndexer.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace Basic\_\_AllSession

{

class S10\_\_ParamsAndIndexer

{

public void Params\_Operation(params int[] number)

{

int sum = 0;

foreach (int i in number)

sum = sum + i;

Console.Write("Addition is: " + sum);

}

public void Params\_Array(params string[] names)

{

Console.Write("\n\nThe Names Are: ");

for (int i = 0; i < names.Length; i++)

{

Console.Write(names[i] + " ");

}

}

public void Params\_MultipleValue(int no1, int no2, int no3, char a, char b, params string[] names)

{

int sum = 0;

sum = no1 + no2 + no3;

Console.Write("\nAddition is: " + sum);

Console.WriteLine($"\nCharacter Are : {a}\t {b}");

Console.Write("\nThe Names Are: ");

foreach (string name in names)

Console.Write(name + " ");

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Indexer \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class Indexer

{

private string[] name = new string[4];

public string this[int index]

{

set

{

name[index] = value;

}

get

{

return name[index];

}

}

}

class IndexerAccept

{

public static void Indexer\_Accept()

{

Indexer index = new Indexer();

index[0] = "Imran";

index[1] = "Shaikh";

index[2] = "Pune";

index[3] = "IPune";

Console.WriteLine("\nFull Name: " +index[0]+" "+index[1]+" "+index[2]);

}

}

}

Program.cs

using Basic\_\_AllSession;

using Basic\_\_AllSession.nestedNamespace; // Assembly Refrence OR Namespace OR PAckage

using System;

namespace Basic\_\_AllSession

{

class Program

{

static void Main(string[] args)

{

//S10\_\_Params

S10\_\_ParamsAndIndexer param = new S10\_\_ParamsAndIndexer();

param.Params\_Operation(20, 22, 33, 44);

param.Params\_Array("Sumit", "Sujit", "Abhilasha", "Imran", "Amit", "Anmol");

param.Params\_MultipleValue(2, 3, 4,'A','B',"Amit", "Somesh","Sohel");

IndexerAccept.Indexer\_Accept();

Console.ReadLine();

}

}

}