

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

How to create and access the class component library (.dll) and consume that file in the console application?

## **Objective:-**

Our objective here is to create and access the class component library(.dll) and then consume it on different console application that the user create.

## **Description:-**

A DLL is a library that contains code and data that can be used by more than one program at the same time. By using a DLL, a program can be modularized into separate components. For example, an accounting program may be sold by module. Each module can be loaded into the main program at run time if that module is installed. Because the modules are separate, the load time of the program is faster. And a module is only loaded when that functionality is requested.

Step1:Create a Class Library that you want.

Step 2:-Write some methods so that the console application can access those methods.

Step 3:-Create a console application containing main method

Step 4:-Go to solution explorer right click on console application and then go to add, then go to references and then add the class library.

Step 5:-Now access the class library in console application using the **using** keyword.

Step 6:-To compile the solution again go to solution explorer and right click on console application and then click on **Set as startup project**.

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ClassLibrary1;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Class1 calc = new Class1();
            calc.sum(10,20);
            calc.subtract(30,10);
            calc.multiply(23,67);
            calc.divide(90,9);
            Console.ReadLine();
        }
    }
}
```

### **Code for Class Library(Class1.css)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ClassLibrary1
{
    public class Class1
    {
        public void sum(int a,int b)
        {
            Console.WriteLine("the sum is:-"+(a + b));
        }
        public void subtract(int a, int b)
        {
            Console.WriteLine("the subtraction is:-"+(a - b));
        }
        public void multiply(int a, int b)
        {
            Console.WriteLine("the multily is:-"+(a * b));
        }
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

```
    }  
    public void divide(int a, int b)  
    {  
        Console.WriteLine("The division is:-"+(a / b));  
    }  
}  
}
```

### **Output:-**

the sum is:-30

the subtraction is:-20

the multily is:-1541

The division is:-10

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

How to give a strong name to an assembly and install that assembly in the global assembly cache?

## **Objective:-**

Creation of strong name to an assembly and then install that assembly to the GAC.

## **Description:-**

An Assembly is a basic building block of .Net Framework applications. It is basically a compiled code that can be executed by the CLR. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An Assembly can be a DLL or exe depending upon the project that we choose.

Assemblies are basically the following two types:

1. Private Assembly
2. Shared Assembly

GAC stands for Global Assembly Cache. It is a memory that is used to store the assemblies that are meant to be used by various applications. GAC contains strong named assemblies. Assemblies in the GAC can be shared by all application running on the machine, without having to copy the assembly locally.

Step1:Go to the Developer Command Prompt for VS code, then run it as administrator.

Step 2:-From there run the command:-

```
C:\Windows\System32>sn.exe -k c:\keypairkey.snk
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Step 3:-then create the library/assemblies you want.

Step 4:-Go to solution explorer, then go to properties then click on it from there go to AssemblyInfo.cs and then write the command :-

```
[assembly: AssemblyKeyFile("C:/keypairkey.snk")]
```

Step 5:- Go to the Developer Command Prompt for VS code, and write the following command to store the assembly in the Global Assembly Cache.

```
Gacutil -I "C:\Users\tansh\source\repos\ClassLibrary1\ClassLibrary1\bin\Debug\ClassLibrary1.dll"
```

## Code:-

### Code for Class Library(Class1.css)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ClassLibrary1
{
    public class Class1
    {
        public void sum(int a,int b)
        {
            Console.WriteLine("the sum is:-"+(a + b));
        }
        public void subtract(int a, int b)
        {
            Console.WriteLine("the subtraction is:-"+(a - b));
        }
        public void multiply(int a, int b)
        {
            Console.WriteLine("the multily is:-"+(a * b));
        }
        public void divide(int a, int b)
        {
            Console.WriteLine("The division is:-"+(a / b));
        }
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

### **Code for the AssemblyInfo.cs:-**

```
[assembly: AssemblyVersion("1.0.0.0")]  
[assembly: AssemblyFileVersion("1.0.0.0")]  
[assembly: AssemblyKeyFile("C:/keypairkey.snk")]
```

### **Output:-**

```
C:\Windows\System32>sn.exe -k c:\\keypairkey.snk
```

Microsoft (R) .NET Framework Strong Name Utility Version 4.0.30319.0

Copyright (c) Microsoft Corporation. All rights reserved.

Key pair written to c:\keypairkey.snk

```
C:\Windows\System32>
```

```
Gacutil -i "C:\Users\tansh\source\repos\ClassLibrary1\ClassLibrary1\bin\Debug\ClassLibrary1.dll"
```

Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.0

Copyright (c) Microsoft Corporation. All rights reserved.

Assembly successfully added to the cache

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

How to disassemble the assembly and regenerate the assembly again?

## **Objective:-**

Assembling and disassembling of assembly using ILDASM and ILASM

## **Description:-**

ILDASM: Intermediate Language Disassembler:-

We use ILDASM.exe to peek at the assembly manifest and IL. You can also use that tool to export manifest and IL to a text file.

ILASM: Intermediate Language Assembler:-

We use ILASM.exe to reconstruct an assembly from a text file that contains manifest and IL.

If I want to feel that how IL looks like are there any tool to disassemble this assembly this can be done by ILDASM

Manifest : consists of meta data about your assembly

Step1:Go to the Developer Command Prompt for VS code, then run it as administrator.

Step 2:-From there run the command:-

```
C:\Windows\System32>ildasm.exe
```

```
C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\ConsoleApp2.exe
```

Step 3:-then a window screen will pop up you can see all the related info about the console application that you created there.

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

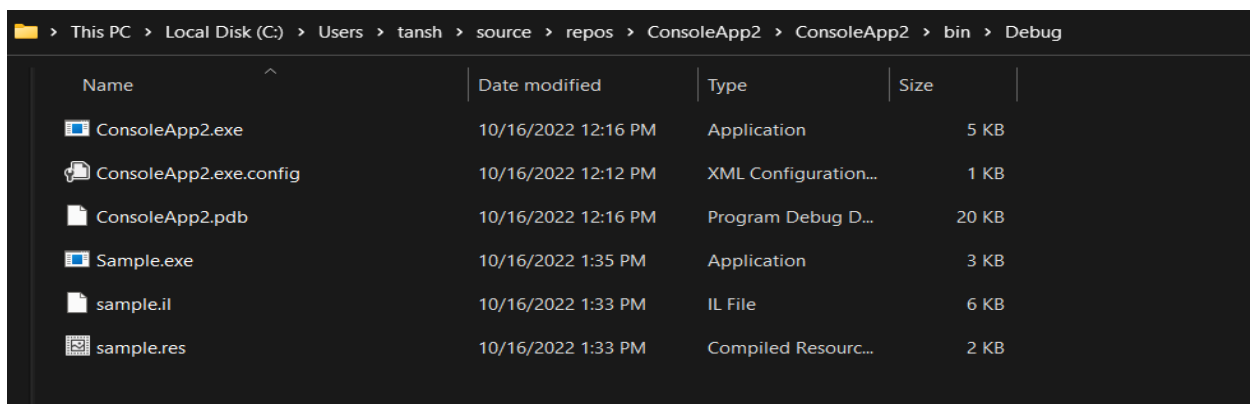


From here go to file and then dump, it will create a Sample.il file for you at the specified path that you gave

Step 4:-Go to the Developer Command Prompt for VS code, then write the command :-

`C:\Windows\System32>ilasm.exe`

`C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\Sample.il`





Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Doing this the Sample.il file will be again converted to Sample.exe.

## **Code:-**

### **Code for Console Application(Program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Naruto Uzumaki v/s Sasuke Uchiha");
            Console.WriteLine("The winner is:-Naruto Uzumaki");
            Console.ReadLine();
        }
    }
}
```

## **Output:-**

C:\Windows\System32>ildasm.exe

C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\ConsoleApp2.exe

C:\Windows\System32>ilasm.exe

C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\Sample.il

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Microsoft (R) .NET Framework IL Assembler. Version 4.8.4161.0

Copyright (c) Microsoft Corporation. All rights reserved.

Assembling 'C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\Sample.il' to  
EXE --> 'C:\Users\tansh\source\repos\ConsoleApp2\ConsoleApp2\bin\Debug\Sample.exe'

Source file is UTF-8

Assembled method ConsoleApp2.Program::Main

Assembled method ConsoleApp2.Program::.ctor

Creating PE file

Emitting classes:

Class 1: ConsoleApp2.Program

Emitting fields and methods:

Global

Class 1 Methods: 2;

Emitting events and properties:

Global

Class 1

Writing PE file

Operation completed successfully

Naruto Uzumaki v/s Sasuke Uchiha

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

The winner is:-Naruto Uzumaki

### **Problem Statement:-**

Demonstrate the usage of namespace by accessing it by:-

- a. Fully qualified name
- b. Using directive
- c. Using alias directive

### **Objective:-**

Accessing the namespace using 3 different methods mentioned above.

### **Description:-**

A **namespace** is designed for providing a way to keep one set of names separate from another. The class names declared in one namespace does not conflict with the same class names declared in another.

Step1: Create a console application and then there create 2 different namespaces having different function name but with same method name

Step 2:-here I have created two namespaces names ConsoleApp3 and ConsoleApp31

Step 3:-Accessing the namespace using the Fully Qualified name, call the method using the following statement:-

```
ConsoleApp3.Anime1.battle();  
ConsoleApp31.Anime2.battle();
```

Step 4:- Accessing the namespace using the Fully Qualified name, call the method using the following statement:-

```
using ConsoleApp3;  
  
using ConsoleApp31;
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Step 5:- Accessing the namespace using the Fully Qualified name, call the method using the following statement:-

```
using CA1B=ConsoleApp3.Anime1;  
using CA2B=ConsoleApp31.Anime2;
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## Code:-

### Code for Console Application(Program.cs) (Using Fully qualified File):-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp3
{
    class Anime1
    {
        public static void battle()
        {
            Console.WriteLine("asta v/s luffy");
            Console.WriteLine("luffy wins");
        }
    }
}

namespace ConsoleApp31
{
    class Anime2
    {
        public static void battle()
        {
            Console.WriteLine("naruto v/s ichigo");
            Console.WriteLine("ichigo wins");
            Console.ReadLine();
        }
    }
}

class Printt
{
    public static void Main(string[] args)
    {
        ConsoleApp3.Anime1.battle();
        ConsoleApp31.Anime2.battle();
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## Code for Console Application(Program.cs) (Using Directive):-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ConsoleApp3;
using ConsoleApp31;

namespace ConsoleApp3
{
    class Anime1
    {
        public static void battle()
        {
            Console.WriteLine("asta v/s luffy");
            Console.WriteLine("luffy wins");
        }
    }
}

namespace ConsoleApp31
{
    class Anime2
    {
        public static void battle()
        {
            Console.WriteLine("naruto v/s ichigo");
            Console.WriteLine("ichigo wins");
            Console.ReadLine();
        }
    }
}

class Printt
{
    public static void Main(string[] args)
    {
        Anime1.battle();
        Anime2.battle();
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## Code for Console Application(Program.cs) (Using alias Directive):-

```
using ConsoleApp3;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CA1B=ConsoleApp3.Anime1;
using CA2B=ConsoleApp31.Anime2;

namespace ConsoleApp3
{
    class Anime1
    {
        public static void battle()
        {
            Console.WriteLine("asta v/s luffy");
            Console.WriteLine("luffy wins");
        }
    }
}

namespace ConsoleApp31
{
    class Anime2
    {
        public static void battle()
        {
            Console.WriteLine("naruto v/s ichigo");
            Console.WriteLine("ichigo wins");
            Console.ReadLine();
        }
    }
}

class Printt
{
    public static void Main(string[] args)
    {
        CA1B.battle();
        CA2B.battle();
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

### **Output 1:-**

asta v/s luffy

luffy wins

naruto v/s ichigo

ichigo wins

### **Output 2:-**

asta v/s luffy

luffy wins

naruto v/s ichigo

ichigo wins

### **Output 2:-**

asta v/s luffy

luffy wins

naruto v/s ichigo

ichigo wins



Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a program in C# to print welcome Message.

## **Objective:-**

Our objective here is to print welcome message using Console.WriteLine() and Console.Write().

## **Description:-**

In C#, to print the data on the console output screen the following method are used –

**Console.Write()** and **Console.WriteLine()** method. **Console** is a predefined class of **System** namespace. While **Write()** and **WriteLine()** both are the Console Class methods.

The only difference between the **Write()** and **WriteLine()** is that Console.Write() is used to print data without printing the new line, while Console.WriteLine() is used to print data along with printing the new line.

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("welcome back....");
            Console.WriteLine("WELCOME BACK");
            Console.Write("how are you");
            Console.ReadLine();
        }
    }
}
```

## **Output:-**

welcome back....WELCOME BACK

how are you

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Demonstrate two ways for writing into console

## **Objective:-**

Our objective here is to print messages using concatenation(+) and placeholder({}) in Console.WriteLine().

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int a, b;
            Console.WriteLine("enter a 1st number=");
            a=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("enter a 2nd number=");
            b = Convert.ToInt32(Console.ReadLine());
            int mul = a * b ;
            Console.WriteLine("the multiplication of\t"+a +"\tand\t"+b +"\tis="+mul);
            Console.WriteLine("the multiplication of {0} and {1} will be ={2}",a,b,mul);
            Console.ReadLine();
        }
    }
}
```

## **Output:-**

enter a 1st number=32

enter a 2nd number=10

the multiplication of 32 and 10 is=320

the multiplication of 32 and 10 will be =320

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a program to demonstrate boxing and unboxing.

## **Objective:-**

Our objective here is to understand the concept of boxing and unboxing in c#.

## **Description:-**

The process of converting a Value Type variable (char, int etc.) to a Reference Type variable (object) is called Boxing. Boxing is an implicit conversion process in which object type (super type) is used. Value Type variables are always stored in Stack memory, while Reference Type variables are stored in Heap memory.

The process of converting a Reference Type variable into a Value Type variable is known as Unboxing. It is an explicit conversion process.

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n= 2020;
            object t1 = n; //boxing
            n = 100;

            System.Console.WriteLine("Value - type value of num n is : {0}", n);
            System.Console.WriteLine("Object - type value of obj is : {0}", t1);
        }
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

```
int n1 = 21;  
int t2 = (int)t1;//unboxing  
System.Console.WriteLine("value of num t2 is : {0}", t2);  
  
Console.ReadLine();  
    }  
}  
}
```

### **Output:-**

Value - type value of num n is : 100

Object - type value of obj is : 2020

value of num t2 is : 2020

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a C# program to check whether the number is prime or not, input will be taken from the user.

## **Objective:-**

Our objective here is to take an input from the user and check whether the number is prime or not.

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int no, f = 0;
            Console.Write("Enter the Number to want to check is Prime: ");
            no = Convert.ToInt32(Console.ReadLine());

            for (int i = 2; i <= no/2; i++)
            {
                if (no % i == 0)
                {
                    Console.WriteLine("Number is not Prime.");
                    f = 1;
                    break;
                }
            }
            if (f == 0)
            {
                Console.WriteLine("Number is Prime.");
                Console.ReadLine();
            }
        }
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

### **Output:-**

Enter the Number to want to check is Prime: 5

Number is Prime.

Enter the Number to want to check is Prime: 6

Number is not Prime.

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a program in C# to check whether a number is even or odd using the ternary operator, input will be taken from the user.

## **Objective:-**

Our objective here is to take an input from the user and using ternary operator check whether the number is even or odd.

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp7
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("enter the number you want to check is even or odd:");
            n = Convert.ToInt32(Console.ReadLine());
            var eveodd=n % 2 == 0 ? "no is even" : "no is odd" ;
            Console.WriteLine(eveodd);
            Console.ReadLine();
        }
    }
}
```

## **Output:-**

enter the number you want to check is even or odd:

34

no is even

enter the number you want to check is even or odd:

55



Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

no is odd

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a program to demonstrate in how many ways you can create a string object.

## **Objective:-**

Our objective here is to see in how many ways can we create a string object.

## **Description:-**

Different Ways for Creating a String:

- Create a string from a literal
- Create a string using concatenation
- Create a string using a constructor
- Create a string using a property or a method
- Create a string using formatting

## **Code:-**

### **Code for Console Application(program.cs)**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp8
{
    internal class Program
    {
        static void Main(string[] args)
        {
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

```
string str = "The tale of Naruto Uzumaki";//the string using literals

string a1 = "naruto";//the string using concatenation operator
string a2 = "fights sasuke";
string a3 = "sasuke wins";

char[] a = { 'a', 's', 't', 'a' };//string using a constructor
string s = new string(a);

string b = "yami sukehiro is a chad person";//string using methods of
the string class
string j = b.Substring(0,30);

string t = "hello";//string using string.Format()
string t1= "anime fans";
string g = string.Format("{0} {1} hope everyone is doing good" ,t,t1);

Console.WriteLine(str);
Console.WriteLine( a1+ a2+ a3);
Console.WriteLine(s);
Console.WriteLine(j);

Console.WriteLine(g);
Console.ReadLine();

    }
}
}
```

## **Output:-**

The tale of Naruto Uzumaki

Naruto fights sasuke sasuke wins

asta

yami sukehiro is a chad person

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Problem Statement:-**

Write a program in C# to demonstrate call by value, call by reference, call by output parameter and call by params

## **Objective:-**

Our objective here is to see the use and working of call by value, call by reference, call by output and call by params

## **Description:-**

Call by value:-

In C#, value-type parameters are that pass a copy of original value to the function rather than reference. It does not modify the original value. A change made in passed value does not alter the actual value. In the following example, we have pass value during function call.

Call by Reference:-

C# provides a **ref** keyword to pass argument as reference-type. It passes reference of arguments to the function rather than copy of original value. The changes in passed values are permanent and **modify** the original variable value.

Call by Output:-

The **out** is a keyword in C# which is used for the passing the arguments to methods as a reference type. It is generally used when a method returns multiple values.

Call by Params:-

By using the params keyword, you can specify a [method parameter](#) that takes a variable number of arguments. The parameter type must be a single-dimensional array.No additional

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

parameters are permitted after the params keyword in a method declaration, and only one params keyword is permitted in a method declaration.

When you call a method with a params parameter, you can pass in:

- A comma-separated list of arguments of the type of the array elements.
- An array of arguments of the specified type.
- No arguments. If you send no arguments, the length of the params list is zero.

## Code:-

### Code for Console Application(program.cs) for Call by value

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp9
{
    internal class Program
    {
        public void print(int n)
        {
            n = n * 10;
            n = n + 20;
            n = n - 10;
            n = n / 10;
            Console.WriteLine("the value within the print function for n will be" + n);
        }
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("enter the value for n");
            n= Convert.ToInt32(Console.ReadLine());
            Program t = new Program();

            Console.WriteLine("value before calling the print function is" + n);
            t.print(n);  

            Console.WriteLine("the value after the print function is called is"+n);
        }
    }
}
```

Name:-Tanishq Singh

Course:-MCA(3<sup>rd</sup> Sem)

Rollno:-75(B)

```
        Console.ReadLine();  
    }  
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## Code for Console Application(program.cs) for Call by refernce

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp9
{
    internal class Program
    {
        public void print(ref int n)
        {
            n = n * 10;
            n = n + 20;
            n = n - 10;
            n = n / 10;
            Console.WriteLine("the value within the print function for n will be" + n);
        }
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("enter the value for n");
            n = Convert.ToInt32(Console.ReadLine());
            Program t = new Program();

            Console.WriteLine("value before calling the print function is" + n);
            t.print(ref n); //cbf

            Console.WriteLine("the value after the print function is called is"+n);
            Console.ReadLine();
        }
    }
}
```

## Code for Console Application(program.cs) for Call by output:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp9
{
    internal class Program
    {
        public void print(out int n)
        {
            int s = int.Parse(Console.ReadLine());
            n = s;

            n = (n * n * n * n * n * n) / 10;
        }
    }
}
```

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

```
        Console.WriteLine("the value within the print function for n will be" + n);
    }
    static void Main(string[] args)
    {
        int n;
        Console.WriteLine("enter the value for n");
        n= Convert.ToInt32(Console.ReadLine());
        Program t = new Program();

        Console.WriteLine("value before calling the print function is" + n);

        t.print(out n);

        Console.WriteLine("the value after the print function is called is"+n);
        Console.ReadLine();
    }
}
```

### Code for Console Application(program.cs) for Call by params:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp9
{
    internal class Program
    {
        public static void Print(params int[] list)
        {
            for (int i = 0; i < list.Length; i++)
            {
                Console.Write(list[i] + " ");
            }
            Console.WriteLine();
        }

        public static void Main(string[] args)
        {
            Print(1, 2, 3, 4);
            int[] myIntArray = { 5, 6, 7, 8, 9 };
            Print(myIntArray);
            Console.ReadLine();
        }
    }
}
```



Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

## **Output:-**

For call by value program:

enter the value for n

5

value before calling the print function is5

the value within the print function for n will be6

the value after the print function is called is5

For call by reference program:

enter the value for n

5

value before calling the print function is5

the value within the print function for n will be6

the value after the print function is called is6

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

For call by output program:

enter the value for n

5

value before calling the print function is 5

10

the value within the print function for n will be 100000

the value after the print function is called is 100000

For call by params program:

1 2 3 4

5 6 7 8 9

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)

Name:-Tanishq Singh  
Course:-MCA(3<sup>rd</sup> Sem)  
Rollno:-75(B)