

OBJECT-ORIENTED PROGRAMMING

C#.NET BASICS2



C#.NET Basics

Functions in C#

- A function is a group of related instructions that performs a specific task.
- Types of Functions in C#:
 1. Built-in Functions
 2. User-Defined Functions
- All the predefined functions in C# are contained limited tasks only i.e. for what purpose function is designed for the same purpose it should be used.
- The User-defined functions in C# are the functions that are created by the programmer so that he/she can use it many times. It reduces the complexity of a big program and optimizes the code.

C#.NET Basics

User-Defined Function in C#

➤ Create

```
<Access Specifier> [Modifier] <Return Type> <Function Name> ([Parameter List])  
{  
    //Function Body / Method Body  
}
```

➤ Example to Create User-Defined Function in C#:

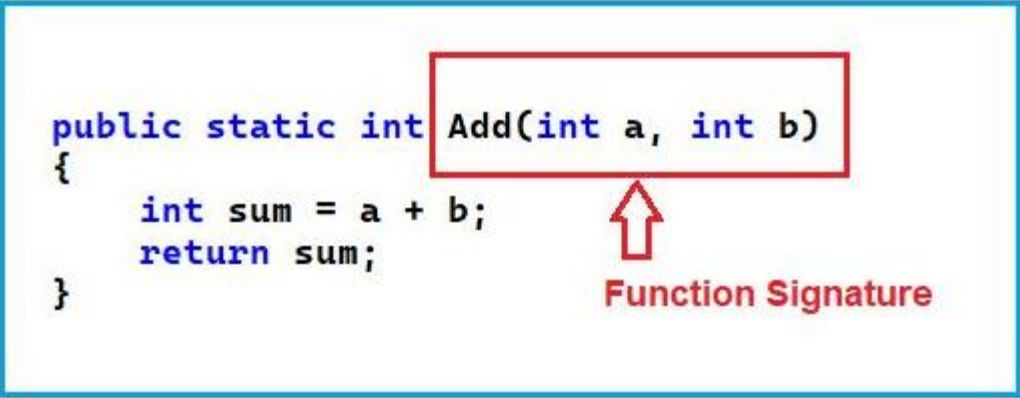
```
public int max(int x, int y)  
{  
    if (x > y)  
        return x;  
    else  
        return y;  
}
```

C#.NET Basics

➤ What is Function Signature in C#?

In C# programming language, a **Method Signature** is consisting of two things i.e. the **Method Name** and the **Parameter List**.

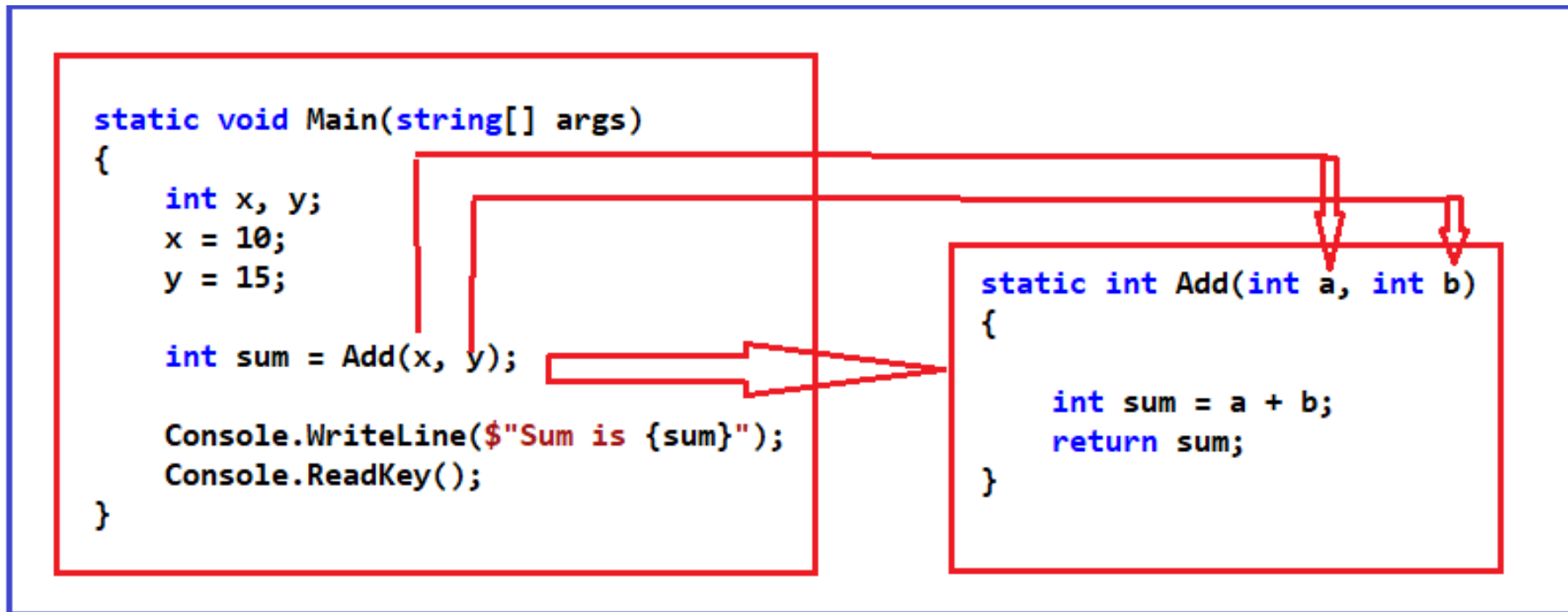
```
public static int Add(int a, int b)
{
    int sum = a + b;
    return sum;
}
```



Function Signature

C#.NET Basics

➤ How to Call a Method in C#?



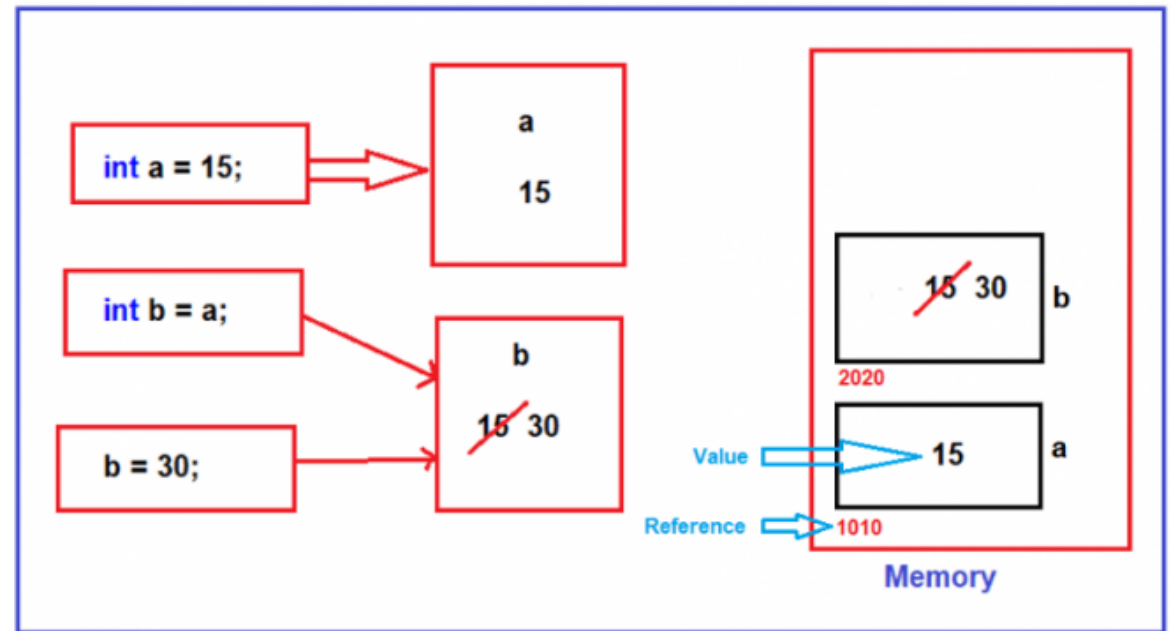
C#.NET Basics

Call by Value and Call by Reference in C#

I. Call By Value in C#

a) Call by value with Variable types in c#

```
static void Main(string[] args)
{
    int a = 15;
    int b = a;
    b = 30;
    Console.WriteLine(a);
    Console.ReadKey();
}
```



C#.NET Basics

- The same thing is also happening when we pass value types to methods.

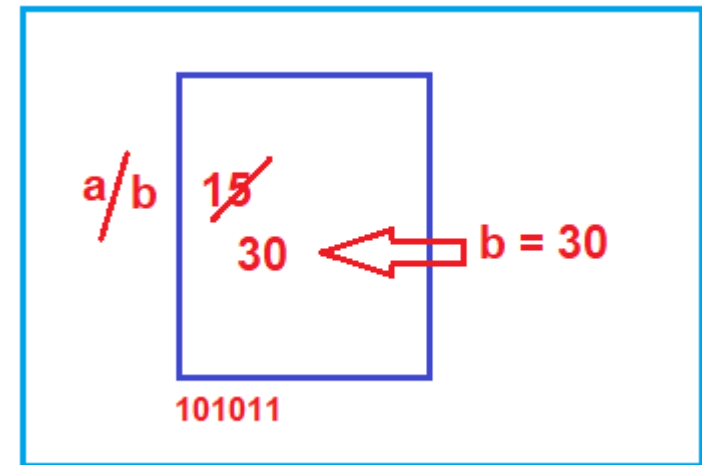
```
static void Main(string[] args)
{
    int a = 15;
    UpdateValue(a);
    Console.WriteLine(a);
    Console.ReadKey();
}
private static void UpdateValue(int b)
{
    b = 30;
}
```

C#.NET Basics

II. Call By Reference in C#

a) Call by Reference with Value Types in C#

```
static void Main(string[] args)
{
    int a = 15;
    UpdateValue(ref a);
    Console.WriteLine(a);
    Console.ReadKey();
}
static void UpdateValue(ref int b)
{
    b = 30;
}
```



C#.NET Basics

➤ Parameter Types

Value parameters

```
static void change(int x)
{
    x = 100;
}
static void Main(string[] args)
{
    int a = 5;
    Console.WriteLine(a);
    change(a);
    Console.WriteLine(a);
}
//out put
//5
//5
```

Reference parameters

```
static void change(ref int x)
{
    x = 100;
}
static void Main(string[] args)
{
    int a = 5;
    Console.WriteLine(a);
    change(ref a);
    Console.WriteLine(a);
}
//output
//5
//100
```

Out parameters

```
using System;
namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 0;
            Program.sum(6, 7, out x);
        }

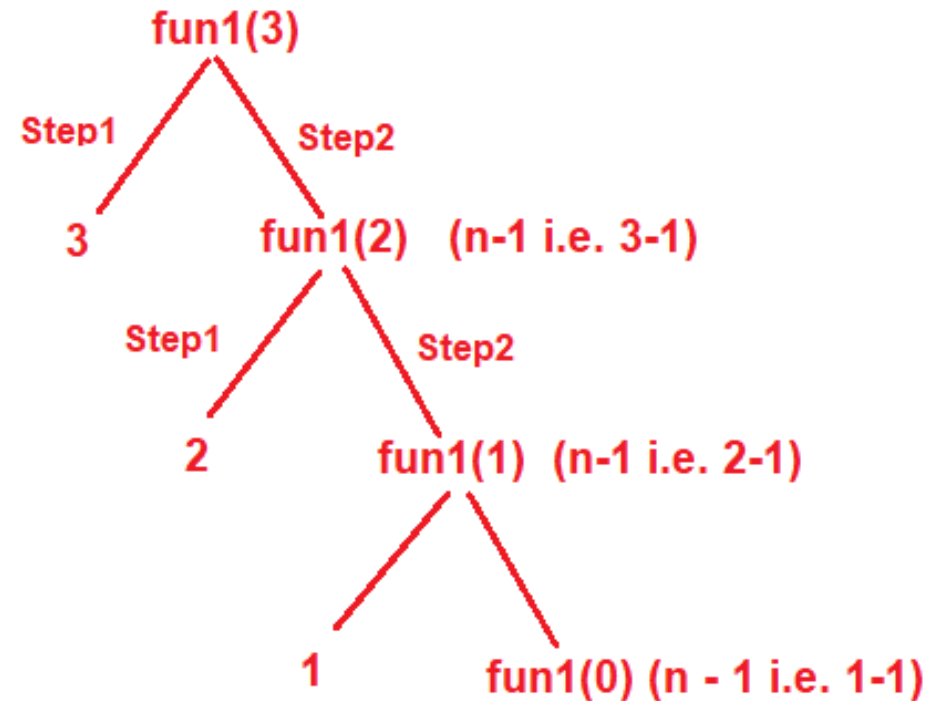
        static void sum(int n1, int n2, out int
s)
        {
            s = n1 + n1;
        }
    }
}
```

C#.NET Basics

➤ Recursion in C# with

```
static void Main(string[] args)
{
    int x = 3;
    fun1(x);
    Console.ReadKey();
}

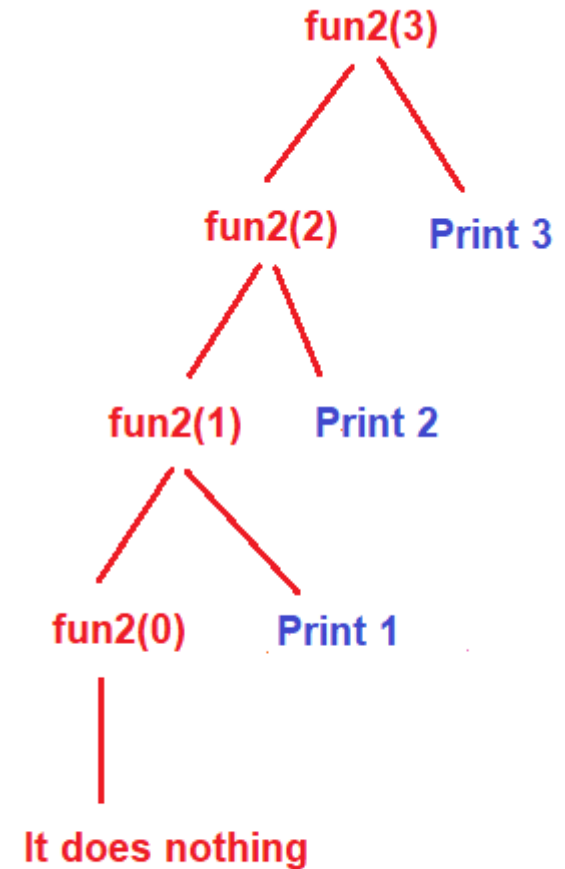
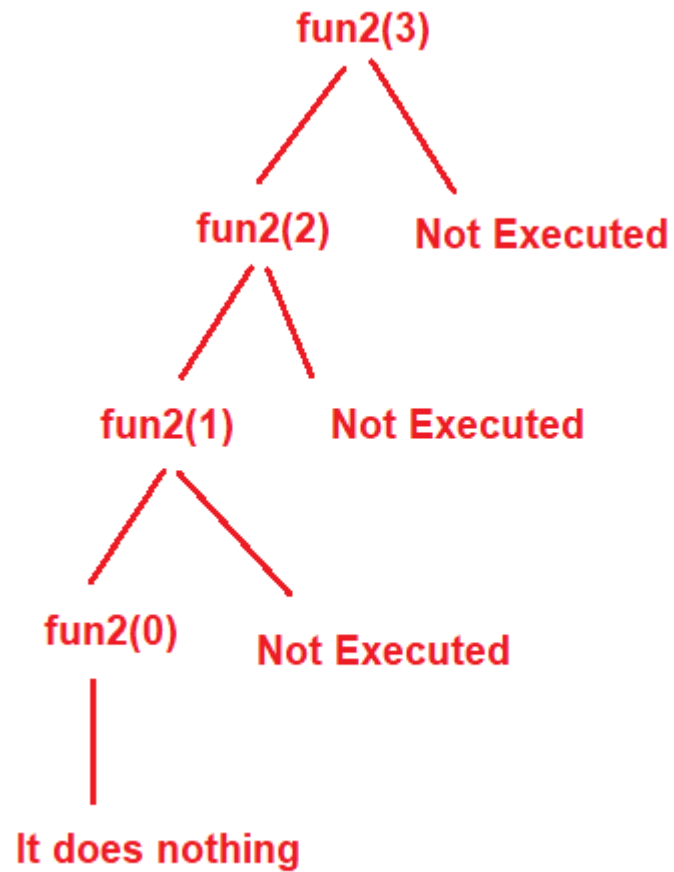
static void fun1(int n)
{
    if (n > 0)
    {
        Console.WriteLine($"{n} "); //1st Statement
        fun1(n - 1); //2nd Statement
    }
}
```



C#.NET Basics

```
static void Main(string[] args)
{
    int x = 3;
    fun2(x);
    Console.ReadKey();
}

static void fun2(int n)
{
    if (n > 0)
    {
        fun2(n - 1);
        Console.Write($"{n} ");
    }
}
```



Output: 1 2 3

C#.NET Basics

String in C#

- In C#, the string is an object of the String class that represents a sequence of characters.
- Strings in C# are reference types i.e. they are not normal data types or you can say they are not like other primitive data types.
- Strings are Immutable in C#.

C#.NET Basics

What are the Differences between String(Capital) vs string(small) in C#?

You can use any one of them i.e. either string or String. But as per the naming convention when you are creating a variable use the small string (i.e. string) and whenever you want to invoke methods on the string then use the capital string (i.e. String) as shown in the below image

```
string str1 = ""; //using small s  
String str2 = ""; //using capital S
```

Use small string to declare variable



```
string str2 = String.Concat(" ");
```

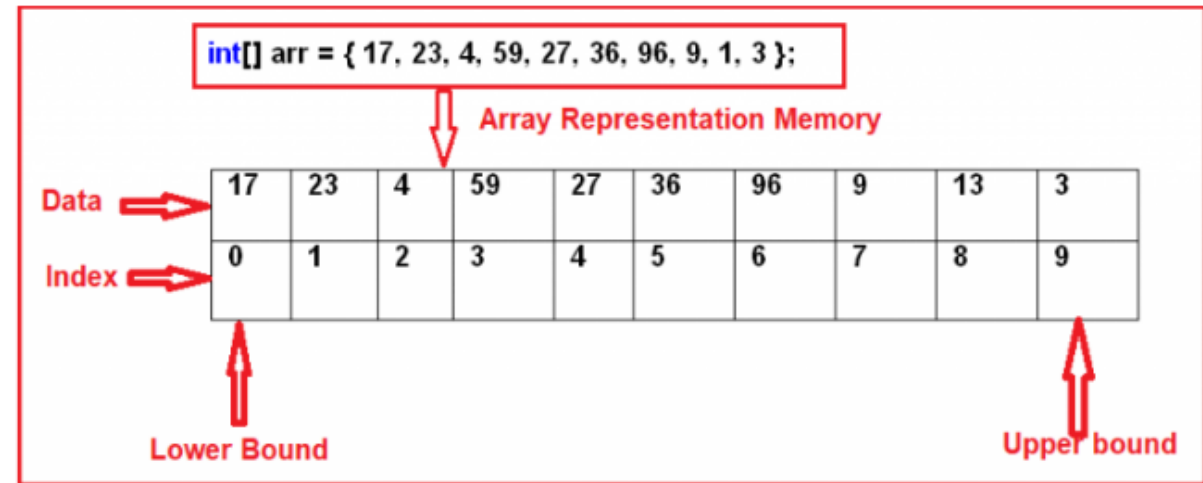


Use Capital String to invoke method

C#.NET Basics

Array in C#

- An array as a collection of similar types of values that are stored in sequential order i.e. they are stored in a contiguous memory location.
- Types of Arrays in C#:
 1. Single dimensional array
 2. Multi-dimensional array
- Memory Representation of Arrays in C#.



C#.NET Basics

One Dimensional Array in C#

The array which stores the data in the form of rows in a sequential order is called a one-dimensional array in C#.

Syntax:

<type>[] <name> = new <type>[size];

Example:

int[] arr = new int[5];

Or

**int[] arr1;
arr1 = new int[5];**

Or

int[] arr2 = {10, 20, 30, 40, 50};

← Initialized using new keyword

← Initialized using argument values

C#.NET Basics

For each loop in C#:

```
static void Main(string[] args)
{
    //Creating an array with size 6
    int[] arr = new int[6];

    //Here we are assigning values to array using for loop
    for (int i = 0; i < 6; i++)
    {
        a += 10;
        arr[i] = a;
    }

    //accessing array values using foreach loop
    foreach (int i in arr)
    { Console.WriteLine(i + " "); }

    Console.ReadKey();
}
```


C#.NET Basics

exercises 2.1

Write a program in C# Sharp to create a user define function.

Expected Output :
Welcome Friends!
Have a nice day!

C#.NET Basics

Exercises 2.2

Write a program in C# sharp to create a function to swap the values of two integer numbers.

C#.NET Basics

Exercises 2.1

- . Write a program in C# Sharp to create a function to calculate the result of raising an integer number to another.

Test Data :

Input Base number: 3

Input the Exponent : 2

Expected Output :

So, the number 3^2 (to the power) 2 = 9