شركــة تحالــف الإمـــارات للحـــلـــول التـقـنـيــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

C# Programming Essential

Tahaluf Training Center 2021

شركــة تحالــف الإمـــارات للحـــلـــول التـقـنـيــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

شركـــة تحالــف الإمـــارات للحـــلـــول التـقـنـيــة ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Day 2

1 **User Input and Numbers**

2 Perform Operators

3 Math Methods

4 String Definition

5 Boolean Definition

User Input and Numbers

The ReadLine() method is a commonly used method or function to take an input from the user until the enter key is pressed. In other words, it is a method that reads each line of string or values from a standard input stream.

User Input and Numbers

In the following example, the user can input his or hers name, which is stored in the variable Name. Then we print the value of Name:

```
namespace MyFirstProject
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter your name:");
            string name = Console.ReadLine();
            Console.WriteLine("Name is: " + name);
        }
    }
}
```

Exercise 1:

Using the previous method, read int value from the keyboard.

User Input and Numbers

Exercise Answer:

The Console.ReadLine() method returns a string. Therefore, you cannot get information from another data type, such as int.

you cannot implicitly convert type 'string' to 'int'.

```
Console.WriteLine("Enter your age:");
int age = Console.ReadLine();
Console.WriteLine("Your age is: " + age);
```

User Input and Numbers

Exercise Answer:

you can convert any type explicitly, by using one of the Convert.To methods:

```
Console.WriteLine("Enter your age:");
int age = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Your age is: " + age);
```

User Input and Numbers

Exercise 2:

Write a Code to read your information like your name, age, …, etc from keyboard, then print it.

User Input and Numbers

Exercise Answer:

```
Console.WriteLine("Enter your name:");
string name = Console.ReadLine();
Console.WriteLine("Enter your age:");
int age = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter your Speciallization:");
string speciallization = Console.ReadLine();
Console.WriteLine("Enter your appreciation:");
char appreciation = Convert.ToChar(Console.ReadLine());
```

User Input and Numbers

Exercise Answer:

```
Console.WriteLine("Your name is: " + name);
Console.WriteLine("Your age is: " + age);
Console.WriteLine("Your Speciallization is: " +
speciallization);
Console.WriteLine("Your appreciation is: " + appreciation);
```

Day 2

1 User Input and Numbers

2 **Perform Operators**

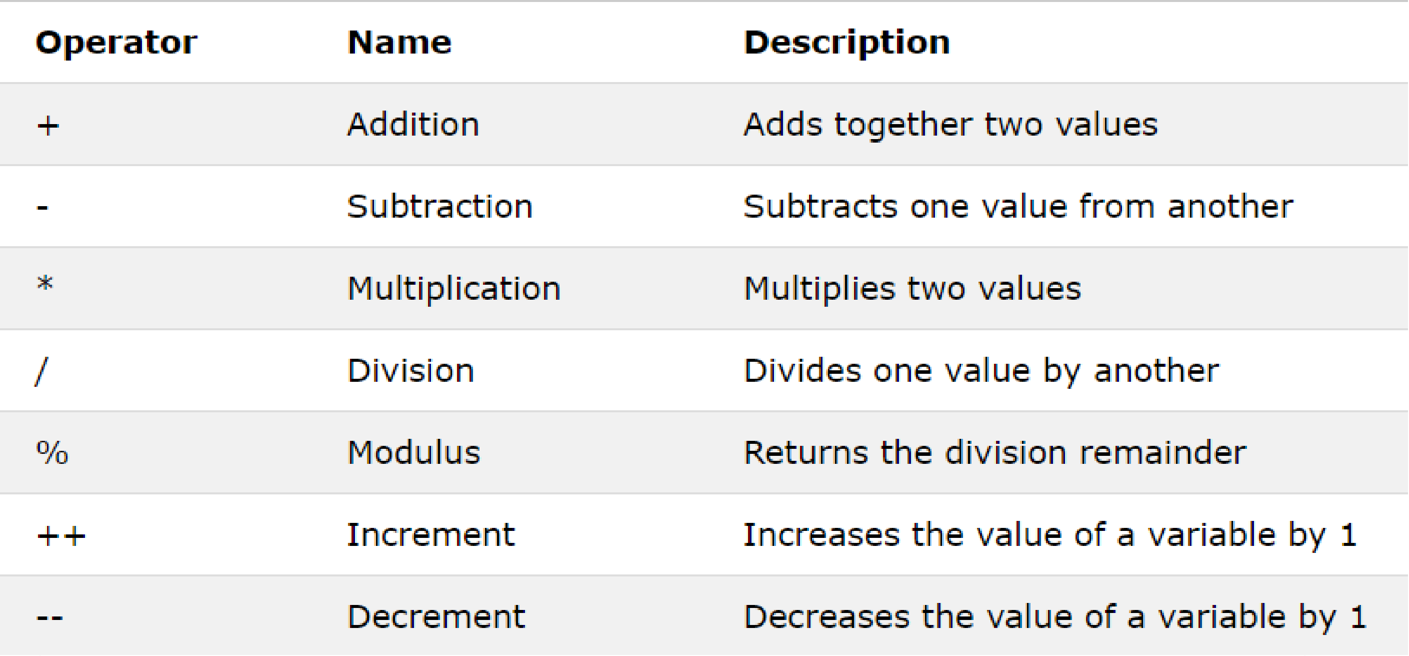
3 Math Methods

4 String Definition

5 Boolean Definition

Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations:



Operator	Name	Description
+	Addition	Adds together two values
-	Subtraction	Subtracts one value from another
*	Multiplication	Multiplies two values
/	Division	Divides one value by another
%	Modulus	Returns the division remainder
++	Increment	Increases the value of a variable by 1
--	Decrement	Decreases the value of a variable by 1

Arithmetic Operators

In the following example, the user can input two integer values, which is stored in the variable Number1 and Number2. Then we print the value of Addition, Multiplication, Subtraction, Division, Modulus, Increment and Decrement:

```
Console.WriteLine("Enter Number1: ");
int number1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter Number2: ");
int number2 = Convert.ToInt32(Console.ReadLine());
int addition = number1 + number2;
int multiplication = number1 * number2;
```

Arithmetic Operators

```
int subtraction = number1 - number2;
double division = number1 / number2;
double modulus = number1 % number2;
number1++;
number2--;
Console.WriteLine("Addition =" + addition);
Console.WriteLine("Multiplication =" + multiplication);
Console.WriteLine("Subtraction =" + subtraction);
Console.WriteLine("Division =" + division);
Console.WriteLine("Modulus =" + modulus);
Console.WriteLine("Number1 increment =" + number1);
Console.WriteLine("Number2 decrement =" + number2);
```

Arithmetic Operators

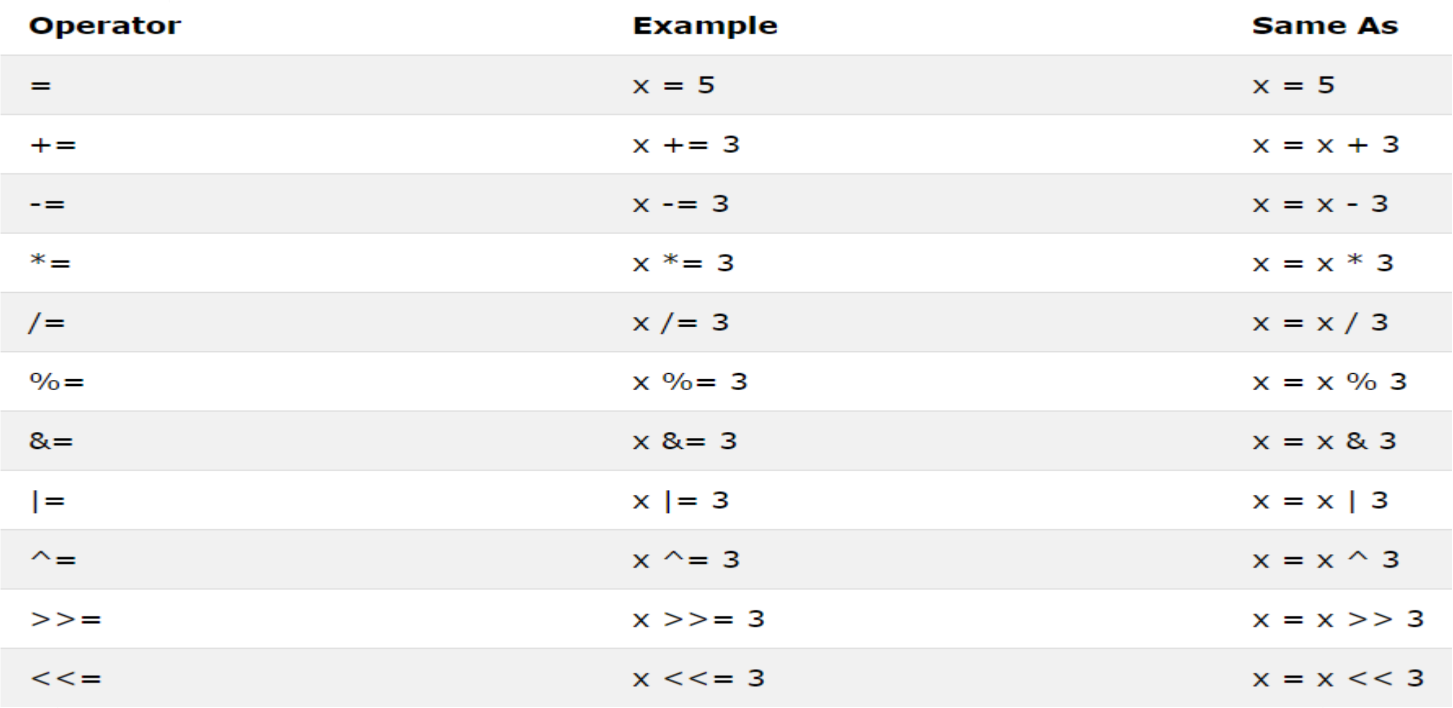
Exercise 3:

Write a C# Sharp program to print the result of the specified operations. Test data:

❑ -1 + 4 * 6
❑ (35+ 5) % 7
❑ 14 + -4 * 6 / 11
❑ 2 + 15 / 6 * 1 - 7 % 2

Assignment Operators

Assignment operators are used to assign values to variables.



Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Exercise 4:

Write a C# Sharp program to input two double numbers, use assignment operators, then print the results.

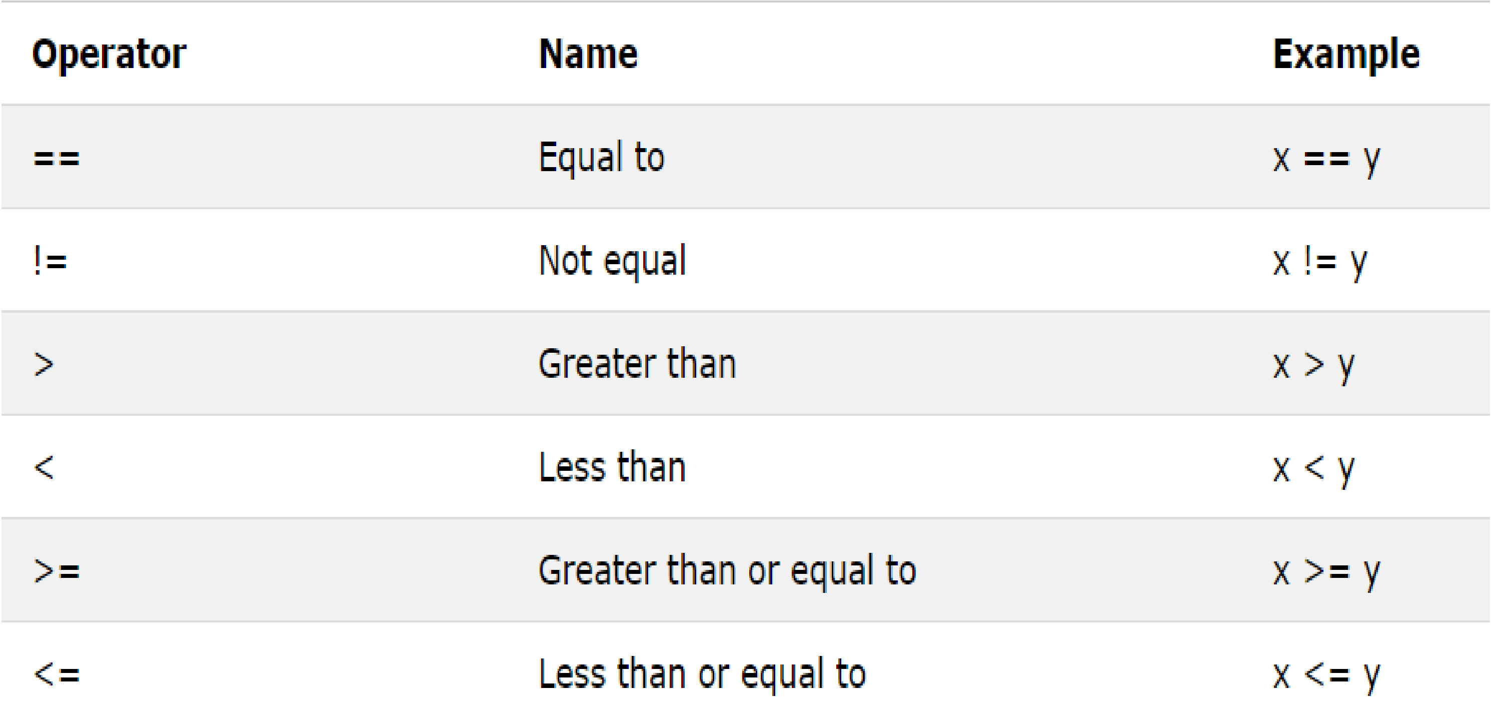
Assignment Operators

Exercise Answer:

```
Console.WriteLine("Enter First Number");
double num1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Enter Second Number");
double num2 = Convert.ToDouble(Console.ReadLine());
num1 += num2;
Console.WriteLine("Add ="+ num1);
num1 -= num2;
Console.WriteLine("Subtraction = "+ num1);
num1 *= num2;
Console.WriteLine("Multiplication="+ num1);
num1 %= num2;
Console.WriteLine("Modulus = "+ num1);
```

Comparison Operators

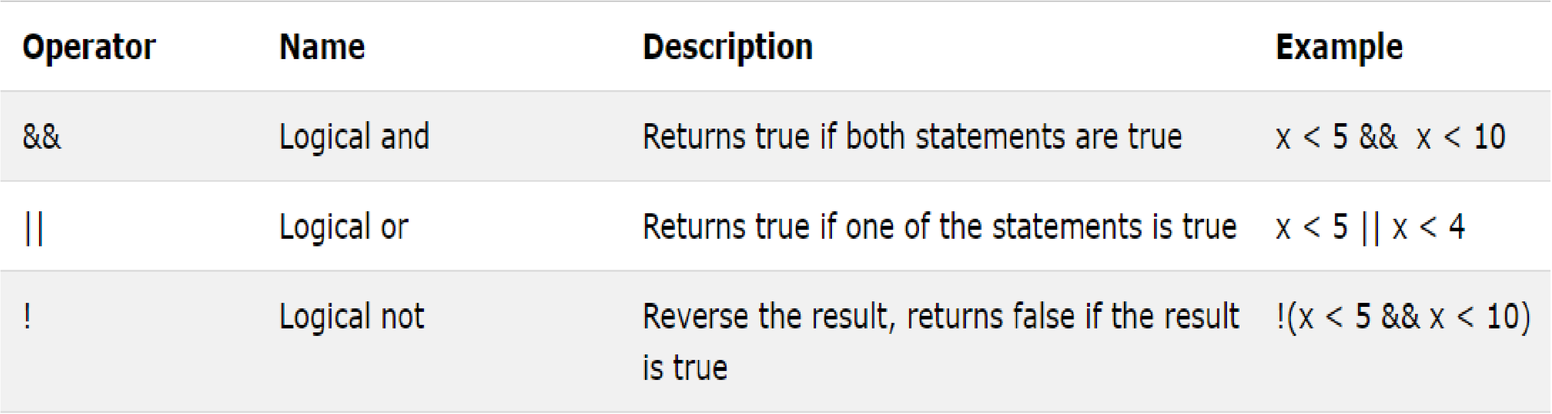
Comparison operators are used to compare two values:



Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Logical Operators

Logical operators are used to determine the logic between variables or values:



Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

Day 2

1 User Input and Numbers

2 Perform Operators

3 **Math Methods**

4 String Definition

5 Boolean Definition

Math Methods

The Math library in C# provides developers with various general, trigonometric, statistical, & logarithmic functions and properties in mathematics.

Exercise 5:

Write a C# Sharp program to input two double numbers one of them is negative, find the highest value and the lowest value, return the square of them, return the absolute of them and round a numbers to the nearest whole number.

Math Methods

Exercise Answer:

```
Console.WriteLine("Enter first number");
double num1 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Enter second number");
double num2 = Convert.ToDouble(Console.ReadLine());
Console.WriteLine(Math.Max(num1, num2));
Console.WriteLine(Math.Min(num1, num2));
Console.WriteLine(Math.Sqrt(num1));
Console.WriteLine(Math.Sqrt(num2));
Console.WriteLine(Math.Abs(num1));
Console.WriteLine(Math.Abs(num2));
Console.WriteLine(Math.Round(num1));
Console.WriteLine(Math.Round(num2));
```

Day 2

1 User Input and Numbers

2 Perform Operators

3 Math Methods

4 **String Definition**

5 Boolean Definition

String Definition

A string variable contains a collection of characters surrounded by double quotes:

```
string coursenum = "12798635421HHHH";

Console.WriteLine(coursenum);
```

String Definition

the length of a string can be found with the Length property:

```
string coursenum = "12798635421HHHH";
Console.WriteLine("The length of the Corse Num string
is: " + coursenum.Length);
```

String Definition

The + operator can be used between strings to combine them. This is called concatenation:

```
string firstname = "Ahmad";
string lastname = "Ali";
string name = firstname + lastname;
Console.WriteLine(name);
```

String Definition

You can also use the string.Concat() method to concatenate two strings:

```
string firstname = "Ahmad";
string lastname = "Ali";
string name = string.Concat(firstname, lastname);
Console.WriteLine(name);
```

Day 2

1 User Input and Numbers

2 Perform Operators

3 Math Methods

4 String Definition

5 **Boolean Definition**

Boolean Definition

A boolean type is declared with the bool keyword and can only take the values true or false.

```
bool isCSharpFun = true;
bool isCSharpNotFun = false;
Console.WriteLine(isCSharpFun);
Console.WriteLine(isCSharpNotFun);
```

Exercise 6:

Write a C# Sharp program to input two int numbers, print true if number1 greater than number2 without using if statement.

Boolean Definition

Exercise Answer:

```
int number1 = 10;
int number2 = 9;
Console.WriteLine(number1 > number2);
```

Any Question?