Lesson 4 - Arithmetic Operations

Comments

They are usually explanatory statements that help the reader understand your code. These comment lines are ignored by the compiler, so comments do not affect your code in any way.

Comments start with a `//` to create a single line comment. Everything in a line after `//` will be treated as a comment. We also have Multi line comments, they start with `/*` and end with `*/` and any number of lines between this will become a comment.

Explicit and Implicit Datatypes – var keyword

Instead of using Explicit datatypes like `int` and `double` to declare a variable, you can also use the `var` keyword and the compiler will determine what should be its datatype based on the value that is being assigned to it. Keep in mind when you initialize a variable using implicit typing, you must assign a value to it because that's the information based on which the compiler can determine the correct datatype for that variable.

Constants

Constants are the opposite of variables, because their value once assigned, cannot be changed. We used the `const` keyword to create a constant. Just like implicitly typed variables, constants must also be initialized with a value on declaration. You cannot change its value once it has been initialized.

Operators

Operators are symbols that perform mathematical or logical operations. Give below is the list of arithmetic operations supported in C#:

OPERATOR	SYMBOL	FORM
ADDITION	+	x + y
SUBTRACTION	-	x - y
MULTIPLICATION	*	x * y

OPERATOR	SYMBOL	FORM
DIVISION	/	x / y
MODULUS	%	x % y

Division – Division by zero, Division between two integers,
Modulus – Remainder operator. Only works on integers.
Operator Precedence – BODMAS, Left to right on equal precedence

Assignment 2

Write a program that takes two integers as input from the user, performs different arithmetic operations on those numbers and then outputs the result as shown in the sample output. Try to use the var keyword to initialize your integers when taking input from user.

Sample Output