C# Basics

INSTRUCTOR: ADEEN BIN WAHEED

LECTURE 4

Arithmetic Operators

Operator	Description	Example
+	Adds two operands	A + B = 30
-	Subtracts second operand from the first	A - B = -10
*	Multiplies both operands	A * B = 200
/	Divides numerator by de-numerator	B / A = 2
%	Modulus Operator and remainder of after an integer division	B % A = 0
++	Increment operator increases integer value by one	A++ = 11
--	Decrement operator decreases integer value by one	A-- = 9

Relational Operators

Operator	Description	Example
==	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

Logical Operators

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non zero then condition becomes true.	(A && B) is false.
||	Called Logical OR Operator. If any of the two operands is non zero then condition becomes true.	(A || B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true.

Decision Making in C#

- if statement:
- if...else statement:
- nested if statements:
- switch statement:
- nested switch statements:

Switch Statement Syntax

```
switch (expression)
{
case value1: statements1
break;
Case value2: statements2
break;
...
Case valuen: statementsn
break;
default: statements
}
```

Switch Statement Example Code

Running Demo

Conditional Operator (Ternary Operator)

Syntax:

expression1 ? expression2 : expression3

If expression1 = true, then the result of the condition is expression2.

Otherwise, the result of the condition is expression3.

Conditional Operator Example

```
if (a > b)
{ max = a;}
 else
{ max = b;}
```

So for the above case there is another alternative way.

```
max = (a > b) ? a : b;
```

Loops

A loop is a sequence of instructions that is continually repeated until a certain condition is reached.

Example:

Formulas used to find average grades for students in a class.

In C# we have following types of loops:

While Loop

Do While Loop

For Loop

Foreach Loop

While Loop

Syntax:

while(expression)

Statement

Example:

```
i = 0;
while(i <= 20)
{
  Console.WriteLine(i + " ");
  i  = i + 5;
}
```

For Loop

Syntax:

for(initial statement; loop condition; update statement)

Statement

Example:

```
for(i = 0; i < 10; i++)
{
  Console.WriteLine(i + " ");
}
```

Arrays

An array stores a fixed-size sequential collection of elements of the same type.

Syntax:

datatype[] arrayName;

arrayName=new datatype[];

Example:

int []arr=new int[10];

Foreach Loop and Array

Running Demo

Escape Sequence Codes

There are certain characters in C# when they are preceded by a backslash. The have special meaning and they are used to represent like newline (\n) or tab (\t). Here, is a list of some of such escape sequence codes:

Escape sequence	Meaning
\\	\ character
\'	' character
\"	" character
\?	? character
\a	Alert or bell
\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\xhh . . .	Hexadecimal number of one or more digits

Escape Sequence Codes Example

Running Demo