Lesson 5 - Assignment and Relational Operators

Assignment Operators

Normally, we have been using the `=` assignment operator up till now, which assigns the value on the right side of the operator to the variable on the left side. There are also **compound assignment operators** in C# that let us perform operations and assignment in a single statement.

Example

This example illustrates the different ways of incrementing a variable by a certain amount.

```
int x = 10;

// without using compound assignment
// equates to x = 15
int y = x + 5
x = y;

// another way to do this without compound assignment
// equates to x = 15
x = x + 5;

// using compound assignment
// equates to x = 15
x += 5;
```

Example

The same shorthand syntax applies to addition, subtraction, multiplication, division and modulus operators as shown in this example.

```
int x = 10;

// addition
x += 5;     // equates to x = 15
```

```
// subtraction
x -= 5;      // equates to x = 5

// multiplication
x *= 5;      // equates to x = 50

// division
x /= 5;      // equates to x = 2

// modulus
x %= 3;      // equates to x = 1
```

Increment Operator

Increment operators have even shorter syntax and increase the value by 1, and this is very commonly used in C#.

```
int x = 10;

// without using compound assignment
// equates to x = 11
x = x + 1;

// using compound assignment
// equates to x = 11
x += 1;

// using increment operator
// equates to x = 11
x++;
```

Decrement Operator

Similarly, we can also use shorthand to decrease the value by 1.

```
int x = 10;

// without using compound assignment
// equates to x = 9
x = x - 1;

// using compound assignment
// equates to x = 9
x -= 1;

// using increment operator
```

```
// equates to x = 9
x--;
```

Relational Operators

Relational Operators are used to evaluate conditions. There are 6 main relational operators. These will return a `bool` type value.

Operator	Description	Example	Result
>=	Greater than or equal to	7 >= 4	True
<=	Less than or equal to	7 <= 4	False
==	Equal to	7 == 4	False
!=	Not equal to	7 != 4	True

Example

```
int age = 25;
bool isAdult = (age >= 18);

// You can also omit brackets and write this as below
// bool isAdult = age >= 18;
// however it looks confusing without the brackets

Console.WriteLine(isAdult);
```

Output

```
True
```