const

- The const keyword stands for constant.
- This means that the variable can be used just as any other variable of its type, but its value cannot be changed. You will get a compiler error if you try to assign a value to a const variable.

Example:

```
const float pi = 3.14;

float x;

// ...

x = pi * 2; // it's fine to use consts in math

pi = 7;      // illegal - you can't write to (modify) a constant
```

Arithmetic Operators

- Arithmetic operators include addition, subtraction, multiplication and division.
- They return the sum, difference, product, or quotient (respectively) of two operands.

```
y = y + 3;

x = x - 7;

i = i * 6;

r = r / 5;
```

The operation is conducted using the data type of the operands, so, for example 9/4 results in 2 instead of 2.25 since 9 and 4 are ints and are incapable of using decimal points.

This also means that the operation can overflow if the result is larger than what can be stored in the data type.

Note: Use the cast operator e.g. (int)myFloat to convert on variable type to another on the fly. For example, i = (int) 3.6 will set i equal to 3.

Compound Assignments

- Compound assignments combine an arithmetic operation with a variable assignment.
- These are commonly found in for loops as described later.
- The most common compound assignments include:

```
x ++       // same as x = x + 1, or increments x by +1

x --       // same as x = x - 1, or increments x by -1
```

```
x += y           // same as x = x + 1, or increments x by +1

x -= y           // same as x = x - 1, or increments x by -1

x *= y           // same as x = x * y, or multiplies x by y

x /= y           // same as x = x / y, or divides x by y
```

Note: For example, x *= 3 would triple the old value of x and re-assign the resulting value to x.

Comparison operators

- Comparisons of one variable or constant against another are often used in if statements to test if a specified condition is true.

```
x  == y          // x is equal to y

x  !=  y              // x is not equal to y

x  <   y              // x is less than y

x  >   y              // x is greater than  y

x  <= y               // x is less than or equal to y

x  >=  y         // x is greater than or equal to y
```

Logical operators

- Logical operators are usually a way to compare two expressions and return a TRUE or FALSE depending on the operator.
- There are Three logical operators, AND, OR, and NOT, that are often used in if statements:

Logical AND:

```
if (x > 0 && x < 5)          // true only if both expression are true
```

Logical OR:

```
if (x > 0 ||  y> 5)           // true only if either expression are true
```

Logical NOT:

```
if (! x > 0)                 // true only if expression are false
```