Types and operators

Integers

```
int numberOfPlatonicSolids;
numberOfPlatonicSolids = 5;
int aHexadecimalNumber = 0xFF;
int anOctalNumber = 013;
int numberOfPlanets = 08; // a bug
short int x;
long int y;
long long z;
unsigned int x;
```

Real numbers

```
float real1;
double real2 = 3.1412;
long double real3;
```

There are some special real values for ∞, not-a-number and -0.0.

casting

Casting means converting types.

```
double x = pow( 1.7815, 0.0 );
int xAsInt = (int)x;
```

Here's another use:

```
int x = 3;
int y = 5;
double fraction = ((double)x)/y;
```

Warnings

Configure the **C/C++** properties of your project to:

- Enable Warnings at level 3.
- Treat warnings as errors.

This will prevent lots of bugs.

Characters

```
char theLetterX = ’x’;
char theLetterY = ’y’;
char theSingleQuoteCharacter = ’\’’;
char theNewLineCharacter = ’\n’;
char* myString = "This is a string";
```

Boolean values

```
bool myBoolean1 = true;
bool myBoolean2 = false;
bool myBoolean3 = (2*2==4); // true
bool myBoolean4 = (2+2>=3); // true
bool myBoolean5 = (2+2==5); // false
```

Note the ==

Operators

```
3 * 5
3 + 5
3 - 5
3 / 5 // gotcha!
3.0 / 5.0
(3 + 5) * 7
5 % 3 // modulus
```

Use `pow` function for powers.

Logical operators

```
(x>0) && (y>0)
(x>0) || (y<0)
true && false // This is false
true || false // This is true
!true // this is false
!false // this is true
```

Use brackets!

Comparison operators

```
1 == 2 // This is false
1 != 2 // This is true
1 > 2  // This is false
1 < 2  // This is true
1 >= 2 // This is false
1 <= 2 // This is true
```

Assignment operators

```
int x = 1;
cout << x << "\n";
```

```
int x = 1;
x += 2;
cout << x << "\n";
```

```
int x = 1;
x++;
cout << x << "\n";
```

Some less common operators

&, |, ~, <<, >> and ^ are used for bit manipulations.

sizeof computes how many bytes of storage are needed.

```
cout << sizeof( char ) << "\n";
char d = ’a’;
cout << sizeof( d ) << "\n";
```