An Introduction to Programming/C++ - Week 1

Ben Goska
Oregon State University

October 11, 2009

Compiler and Editor

The main tool that is used for C++ development is the compiler. There are many avaliable, but we will use g++. To install g++:

- Windows: Download and install Dev-C++
- Linux: You should have g++, if not install it with the package manager

For editing code one of the following editors is recommended.

- gedit - windows + linux
- scite - windows + linux
- vi,vim,gvim - windows + linux
- notepad++ - windows

Compling

To compile a program using g++ you usually use the command line with a command similar to:
g++ file.cpp -o myProg
which would compile file.cpp into a program called myProg

Writing down what we want to happen

Pseudocode: Writing down exactly what you want to happen

Example of pseudocode to make peanut and jelly sandwich

```
1  Get bread, peanut butter, jelly
2  put peanut butter on bread
3  put jelly on bread
```

Types of Variables

Variables can be used to store things, this includes strings (text), whole numbers, decimal numbers, or even arrays (lists) of numbers.
Example of pseudocode to do a little math $10\frac{2}{14} + 6$

```
1  result = 0;
2  result = result + 10;
3  result = result * 2;
4  result = result / 14;
5  result = result + 6;
```

Loops and things

Loops are used to do something repeatedly, there are many types of loops. The most important kind of loop is a simple "while" loop. This type of loop runs while some condition is true, this is best shown in an example, here is an example of a loop to add up the numbers from 1 to 10.

```
1  sum = 0;                  //start the sum at 0
2  i = 1;                    //start counting at 1
3  while (i <= 10) {          //loop until i > 10
4       sum = sum + i;
5       i = i + 1;
6  }
```

Pseudocode to C++

Moving from being able to write pseudocode to writing C++ is not too difficult, the main thing to remember is that C++ is very picky about how you write things.
As a good rule you should start by writing pseudocode for what you want to accompilsh.

Hello World

This is the most basic, but complete, C++ program.

```
1 #include <iostream>
2
3 int main(void) {
4     std::cout << "Hello World" << std::endl;
5     return 0;
6 }
```

Hello World, explained

Line 1: This line includes the standard functions for outputting text to the screen.
Line 3: This is the basic definition of a program, this is where code starts to be executed.
Line 4: This puts the string "Hello World" into the output "cout", or writes "Hello World" to the console.
Line 5: The program ends on this line, returning the value "0" to the operating system (this indicates no failure occured).

Types of variables

A few basic types in C++

- int - Integer, holds whole numbers (32-bit)
- char - character, holds one character (8-bit)
- double - Floating point, holds decimal numbers
- std::string - String, holds text

Modifiers that can be applied to variables

- unsigned - only holds non-negative numbers
- volatile - can change at any time (shouldn't ever be needed in normal code)

Declaring variables

When you declare a variable you must include the following things:

- type
- name

And you may or may not have

- modifiers
- initial value

Declaring variables

Example: declare an unsigned integer, with an initial value of 0

```
1 unsigned int myInteger = 0;
```

Note that variable names cannot have spaces, or special characters. So you may use any letters (A-Z, a-z), numbers (0 - 9).

Using variables

All variables have operators that can be used, for numbers these are the familiar operations, that include:

- $+$ Addition
- $-$ Subtraction
- $*$ Multiplication
- / Division (NOTE: watch out for rounding)
- % Modulo (remainder of division)
- $=$ Assignment

Using variables

Now we can put all these together however we want to, including the use of parentheses, an example that expands on our our hello world program.

```
1  #include <iostream>
2
3  int main(void) {
4      int i = 0;  //make a variable to work with
5      i = (i + 10) * 2;
6      std::cout << i << std::endl;
7      i = i - 2;
8      std::cout << i << std::endl;
9      i = i / 2;
10     std::cout << i << std::endl;
11     return 0;
12 }
```

Types of loops

C++ offers a number of different loop types, really they all perform the same function, though some are better for certain situations.

- while - The while loop is the simplest loop, it does something while a condition is true
- for - The for loop is just like a while loop but can initialize a variable and perform specific actions each loop

The while loop

This program prints out the numbers 1 to 10

```
1  #include <iostream>
2
3  int main(void) {
4      int i = 1;
5      while (i <= 10) {
6          std::cout << i << std::endl;
7          i++; //short hand for i = i + 1;
8      }
9      return 0;
10 }
```

The for loop

This for loop does that same as the previous while loop.

```
1 #include <iostream>
2
3 int main(void) {
4     for(int i = 1; i <= 10; i++) {
5         std::cout << i << std::endl;
6     }
7     return 0;
8 }
```

The if statement

The if statement is perhapse one of most useful statements in any programming language.

```
1  #include <iosteam>
2
3  int main(void) {
4      for(int i = 1; i <= 100; i++) {
5          if (i % 3 == 0) {
6              std::cout << i << std::endl;
7          }
8      }
9      return 0;
10 }
```

Things to work on now

A few quick things to try out now

- Make a loop that adds up the numbers between 1 and 100, print the result
- Make a loop that adds up integers starting with 1 until the result is greater than 100