

## Introduction to C++

(And tools you'll need this semester)  
9-7-2001

---

---

---

---

---

---

---

---

## Opening Discussion

- Take first quiz.
- What did we talk about last class?
- Where you able to get your CS accounts set up? If you did them try logging into them now.
- Are you at all familiar with the command line interface?

---

---

---

---

---

---

---

---

## Logging In

- One advantage of Linux is that you can access the machines remotely.
- Putty is a utility that lets you do a secure login to the CS accounts from a windows machine. I've put a link to it on the on the main course page.
- The CS machines are behind firewalls so if you are logging in from a computer that isn't associated with Trinity you might have problems.

---

---

---

---

---

---

---

---

## vi

- To write a program all you really need is a text editor. For this class I recommend vi as a text editor.
- You might find gvim to be closer to what the type of system you would want to use. It also has nice color coding.
- You can customize what vi does using the .exrc file in your home directory.

---

---

---

---

---

---

---

---

## Simple Program

- Here is a simple program in C++.

```
// A simple example program.
#include <string>
#include <iostream>

void main() {
    string name;
    cout << "What is your name? "; // Prompt
    cin >> name; // Read the name in.
    cout << "Hello " << name << endl;
}
```

---

---

---

---

---

---

---

---

## Tokens

- The fundamental unit in a C++ program is the token. C++ in general ignores white space between tokens, but white space in a token will change its meaning.
- In the sample program, examples of tokens would be
  - | main
  - | ;
  - | cin
  - | "Hello "

---

---

---

---

---

---

---

---

## Expressions

- Tokens can be put together in specific way to form expressions. There are different types of expressions in C++ that we will talk about over the next few weeks. Monday we will discuss numeric expressions.
- An example numeric expression would be "num+3" which contains the tokens "num", "+", and "3".

---

---

---

---

---

---

---

---

## Statements

- The next level up in the C++ language is the statement. In some ways a statement can be thought of as a single thought or directive. This is not always the case, but it is a valid way for you to think about it at this point.
- More accurately, a statement is a section of code ending with a ";".
- It has the value of the last expression evaluated in it.

---

---

---

---

---

---

---

---

## Blocks

- Statements can be grouped together to form blocks using curly braces: { and }.
- This produces what is called a block of code. The book also calls this a compound statement.
- In our example there is only one block for the main function.
- A block has the value of the last statement evaluated in it.

---

---

---

---

---

---

---

---

## Comments

- So far we have been talking about things that will wind up being part of the instructions for the computer to perform a task. There is another very important part of your programs that the compiler completely ignores: comments.
- Comments in C++ come in two forms and should be used liberally to indicate what is happening in the program.
  - // The rest of the line is a comment.
  - /\* The stuff between is a comment (could span lines). \*/

---

---

---

---

---

---

---

---

## Compiler Directives

- At the top of the sample program were the lines `"#include <string>"` and `"#include <iostream>"`. These lines are used to say that the program will be using pieces of code found in those libraries that came with the compiler.
- Later in the semester we might use the form
  - `#include "mylib.h"` which includes a file in the same directory as the source code.

---

---

---

---

---

---

---

---

## Main

- Every program written in C++ starts in a function called main. We will talk more about functions in a week.
- For now we will just say that every program needs one function called main and that the code for a function is a code block (multiple statements encased in curly braces).

---

---

---

---

---

---

---

---

## Streams

- In C++ the standard method of writing things to screen or reading a users keyboard input is streams. The sample program used the two main predefined streams: cin and cout.
  - cin is the standard input stream. The input operator (>>) tells the computer to move data from input into a variable.
  - Cout is the standard output stream. The output operator (<<) causes values to print on the screen.

---

---

---

---

---

---

---

---

## g++

- Now we will go into a Linux shell and use g++ to compile our sample program.
- To do this we need to be in the directory where the source file is located and enter the command "g++ test.cpp". Here "test.cpp" is the name I gave to my source file. You might call yours something different.
- This produces a file called a.out that is executable (we can type that in at the prompt and the computer can understand what the file says).

---

---

---

---

---

---

---

---

## Minute Essay

- Today we quickly went through some of the things that you will need to know to be able to complete assignments. It was a quick treatment and you should have questions about how to do things. Is there anything you really want me to touch on Monday?
- This page has a tutorial to help you get around in Unix/Linux.
  - <http://www.cs.trinity.edu/~thicks/UNIX/Unix.html>

---

---

---

---

---

---

---

---