

# Lecture 2 Review

.login and .tcshrc (especially) are important home directory files.

Relevant only for TC shell (which is what we are using).

Know how to set variables.

Distinguish quotation marks: “ v. ‘ v. `

Output redirection: > or >> or >!

gedit is a simple-to-use text editor.

enscript and lpr used for printing. Know your printer queue name.

➤ Read (and perform!) Linux tutorial

**!! Scream if you get stuck !!**

# Our First C++ Program

Time to get dangerous: write/run a program

Write the following text in a file named `hello.cc`

```
// First C++ program, maybe
#include <iostream>
int main ()
{

    // write greeting

    std::cout << "Hello, World!" << std::endl;

    return 0;
}
```



Note file extension

**!! Scream if you get stuck !!**

# Hello World

// First C++ program, maybe

#include <iostream>

int main ()

{

// write greeting

std::cout << "Hello, World!" << std::endl;

return 0;

}

#include directives. Std libraries v. core language  
iostream (io ~ input/output) is part of std C++ library,

Formulaic beginning. int stands for integer  
main is *function* name. produces ("returns") an integer, hence the int. Takes no arguments so the () is empty. Functions can have different names and return other than integer values.

// comment indicator. Characters to right ignored

std::cout is standard output "stream." The std says that the cout stream in the standard part of the C++ library is what you want to use. Could be others. std::endl is equivalent to the enter key on your keyboard. The ; indicates the end of a line of programming instructions. DO NOT FORGET THIS.

Says to return the integer 0. NOTE THE ; SIGN. DON'T FORGET IT.

Needed

Needed

# Hello World: More Detail

```
// First C++ program, maybe
```

Again, a comment. No effect on program execution. Short note to reader.

```
#include <iostream>
```

Leading pound sign (#) indicates “preprocessor directives,” not regular code lines w/ expressions. We need this part of the standard C++ library so we include it. One such include statement per line.

```
int main ()
```

```
{
```

Start of the definition of the main function, the point where all C++ programs begin their execution. True even if file contains functions before and after main. All C++ programs must have a main function.

```
// write greeting
```

```
⋮
```

Req'd. Everything between this and its closer is what main does.

# Hello World: Yet More Detail

C++ statement. A statement is an expression that produces some noticeable effect. **All** statements terminated with a ;

```
std::cout << "Hello, World!" << std::endl;
```

```
return 0;
```

std::cout represents standard output stream in C++  
“Declared” in standard iostream file (see previous page).

The << should remind you of redirection.

The std::endl is not really necessary.

Equivalent to “enter” key on keyboard.

Req'd. Causes main to finish. A return of 0 indicates good news.

Req'd. Terminates main function.

# Help with C++ Program Structure

My head is exploding.



I need something to read quietly, at my own pace.

[http://www.cplusplus.com/doc/tutorial/program\\_structure.html](http://www.cplusplus.com/doc/tutorial/program_structure.html)

Link also available from PHYS 3340 links page

# Simple Program Compilation & Execution

We need to “compile” (say what?) the program we just wrote.

“**compile**” means turn the human readable commands  
Into machine readable commands (loosely speaking).

g++ -o hello hello.cc

compiler “switch.” o means next file name is your name for the  
the “executable” file. Name it what you want.

file that contains “source” code.

particular compiler name (your machine has several)

Q: cat hello.cc

example of “ascii” file

Q: cat hello

Yuck. example of “binary” file

Q: hello

Shiver me timbers! Your first C++ program.

# String Input

```
// ask for a name
#include <iostream>
#include <string>

int main ()
{
    std::cout << "please enter a first name: ";
    // read the name
    std::string name;
    std::cin >> name;
    // write greeting

    std::cout << "hello, " << name << "!" << std::endl;

    return 0;
}
```

Just a single-line comment.

Need another piece of std library (to deal w/ “strings”)

We omitted the `std::endl` character this time. OK.

Note use of quotation marks. Note `;` character.

Declare name to be a variable of type `std::string`

Executable code and comments on same line OK.

Want to read in from std input (aka keyboard)

Catenate “string literals” and variables OK.

Seen before.



# Simple Program Compilation & Execution (2)

Compile the previous program. Pick your own file names:

`g++ -o howdy greetings.cc` (These are mine. Choose your own.)

Q: execute howdy (or whatever it was you called the executable file.

Q: What do you notice about the first output line? Where does it end?

Q: Can you guess why? (i.e., how did you tell the computer to do this?)

Q: Modify your greetings.cc to, say, compliment you, insult you, me....)

**!! Scream if you get stuck !!**

# Summary

C++ programs have a certain structure.

`#include <"standard header">` (I will tell you these.)

`//` indicates a comment. (There is also another way.)

The `main` function is vital to any C++ program.

`std::cout` and outputting characters to the screen.

`g++` is our compiler command

Finish linux tutorial

**Don't suffer in silence. Scream for help!!!**

