*9/29 Lecture Notes*

History of C++

Most programming languages are just used by the people who invent them – only very few are used by the majority of programmers (~10)

Different languages are used for different purposes like business, design, etc.

Design goals of C++: compile to efficient code, high performance on runtime; tradeoff is less built-in checks for mistakes so code errors can fuck everything up

1971: Dennis Ritchie from Bell Labs invents new language C; made to be fast

Used within Bell Labs only until 1975/1976 when they decided to make the UNIX OS and C available for very cheap to universities. Students learn C, graduate, and spread it to businesses and corporations when they enter the workplace.

C had issues – hard to change certain parts, compiler didn’t catch as many errors as it could. In 1981 Bjarne Stroustrup developed “C with classes” or C++

By 1985 C++ was released from Bell Labs and it caught on fairly quickly. By 1990 there was a substantial body of C++ programs and compilers available.

Took until 1998 for ISO to release C++ Standard, a universal version of the language. Took a while for many compilers to catch on; one notable exception was Microsoft because they didn’t want people porting Windows applications to other machines.

ISO is currently revising the standard and it is expected that by 2012 a revised version of the standard will be released.

Writing C++

int main()  
{  
 cout <<”Hello!” << endl;  
}  
  
*cout:* Anything written to cout (see-out) goes to the screen. The “<<s” are operators; they indicate “pointing” the string towards cout.

*endl:* Basically says “move the cursor to the next line”

Statements end in semicolons to indicate the end of the line.

The structure of the code doesn’t matter to the compiler but makes it easier to read for humans who look over your code. C++ is a case sensitive language; if you begin with int Main() instead of int main() the program will fail. All key words such as if, int, while, return are lower case.

Every name that you use in C++ has to have previously been defined. The previous bit of code would *not* compile because cout and endl have not been defined yet – the compiler doesn’t know what they are.

#include <iostream>  
using namespace std;  
  
int main()  
{  
 cout <<”Hello!” << endl;  
}

Throw in which namespace you are using – in this case standard or std. Sometimes you might load other libraries and use different namespaces; in this case you can also refer to names with the library and 2 colons in front of the name, like a last name to identify. Ex: std::cout or std::endl.

(Refer to “class sample 9\_29”)

*cin* is the standard input source. Syntax is *cin >> variable.*

You have to announce variables or *declare* them before you use them. Declare variables as a type -- in the case of hoursWorked, use *double* because you want to store a number with a decimal.

When declaring variables, you have to follow certain rules: has to start with a lowercase letter, can use letters, digits, or underscores. Convention is toSpellDifferentWordsLikeThis, uppercasing new words while keeping the first letter lowercase.