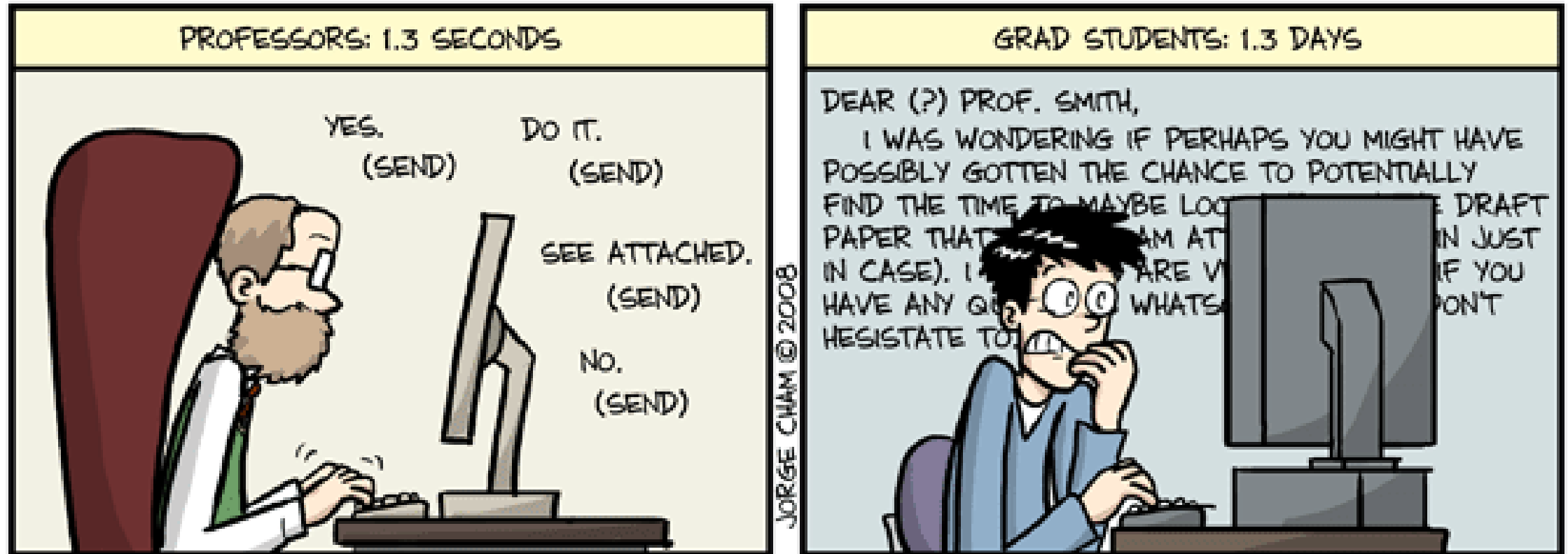


# CS 161, Lecture 1: C++, Data Types, Variables, Constants – 10 January 2017

## AVERAGE TIME SPENT COMPOSING ONE E-MAIL



# C++

- High level language -> not binary
- Language is compiled into binary for the computer to understand
- How to do this conversation:
  - Create cpp file -> vim my\_file.cpp
  - Add some code to file (basic boiler plate)

```
#include <iostream>
using namespace std;
int main() {
    //my awesome code goes here
    cout << "Line of code that prints this line to the screen." << endl;
    return 0;
}
```

# Running Your Program Continued

- Save your file -> <Esc> :wq
- Compile your program -> `g++ my_file.cpp -o my_file_exe`
- Run your program -> `my_file_exe`

# Break Down of the Boiler Plate

```
#include <iostream>
using namespace std;
int main() {
    //my awesome code goes here
    cout << "Line of code that prints this line to the screen." << endl;
    return 0;
}
```

Library: imports functions, macros, etc. other people wrote to take care of things for you

Namespace: provides a grouping for identifiers and prevents name conflicts with libraries

Function Main: the function the g++ compiler looks for to start your program

// this is a comment, this is not read by the compiler

the rest is awesome code contained in the function

# More About Comments

- `//` single line comment
- `/*` Block comment, can be spread across  
Multiple lines `*/`
- Why comment?
  - To help you outline/mark your code when developing (remove these later)
  - To help others understand particularly complex pieces of code
  - To orient someone to your program/code

# Commenting and Style for this Class

- There is a style guide -> find it on the website
- For now you should include program headers

```
/******  
** Program: my_file.cpp  
** Author: Shannon Ernst  
** Description: example boiler plate which prints a line  
** Input: None  
** Output: text to screen  
*****/
```

# String Literals and Escape Sequences

- String literals are denoted with quotes `""`
  - Correct -> `cout << "Hello World" << endl;`
  - Incorrect -> `cout << 'Hello World' << endl;`
  - Incorrect -> `cout << "Hello  
World" << endl;`
- Escape characters to display special characters
  - Denoted with `\`
  - Example newline: `cout << "Hello World \n";`

# Data Type

- Data: information -> literal, variable, file, etc.
- Data type: description of the kind of information
  - Primitive: Int, Float, Double, Long, Short, Char, Bool
  - User defined: objects, classes -> dealt with in CS 162
- Basics:
  - int: whole numbers ex: 42, -7, 0
  - double: real numbers ex: 3.14, -237.15, 6.0221409e+23
  - char: characters ex: 'A', '!', 'f', '\'
- Signed (negative and positive), Unsigned (positive including zero)



# Variables

- Location in memory
  - Has name – can be anything ex: my\_num, bob, x, horriblyLong\_and\_bad
  - Has type – see data types, indicates how much space needs to be carved out
- Declaration
  - Statement requesting that memory be carved and named accordingly
  - Ex:
    - `int number_of_students;`
    - `char letter_grade;`

# How to Name

- Names also known as identifiers are given to variables and function
- Start with letter: upper case, lower case, underscore
- Followed by sequence of letters and digits
  - Good: myVar, result\_of\_eq1, \_hello
  - Bad: 1234, my-Var, 2eq\_res
- Can't use keywords

# Assigning Values to Variables

- Point of variables is to hold data
- Declare a variable
  - `int my_num;`
- Use '=' followed by the data you want to store (data must be same type as what was declared)
  - `my_num = 5;`
- '=' is the assignment operator not a test for equivalence
  - say my\_num "is assigned" or "gets" 5
- Can declare and assign on same line
  - `int my_num = 5;`

# Printing Variables

- `cout << "The result is: " << result << endl;`
- Alters out stream

# Constants

- Constants do not change
- Two ways to create a constant
  - Define a macro
    - At top of program, no semicolon
    - `#define MAX_SIZE 10000`
    - MAX\_SIZE will always be 10000 through out the entire program
  - Use const keyword
    - Same as declaring variable
    - `const int MAX_SIZE 10000;`

# Predefined Macros

- Some macros already exist for things, typically import in library
- C++: <limits>
- Use MIN and MAX