CS 161
Intro to CS I

Variables, Constants, Expressions, and User Input
Odds and Ends

• Assignment 1 due tonight!!!
• Please make sure you have a way to get information onto/off the ENGR server.
  – Map a network drive
  – Transfer files: Filezilla
• Assignments must compile and run on ENGR!
• Demos start this week (no laptop required).
• Sign-up for demo on home page, after you submit your assignment.
More C++

- Programming Style: please read your class style guide
  - Program Header/Description
  - Placement of {}
  - Indentation: spaces vs. tabs
- String Literals (Strings)
  - Quotation marks not single quotes!
    - INCORRECT: std::cout << ‘Hello World’;
  - Do not span more than one line!
    - INCORRECT: std::cout << “Hello World”;

OSU Oregon State University
More C++

- Escape Sequences
  - Display special characters
  - Use backslash, \\, before special character to print
- Examples:
  std::cout << "\"Hello World\\"\n";
- Refer to book for common escape sequences.
Comments

• Ignored by compiler
• Comment a block of code: /*…..*/
• Comment one line of code: //
• Why use these?
• What are you required to have right now?
  – Header at beginning of program
    /*************************************************************************/
    ** Program: hello.cpp
    ** Author: Jennifer Parham-Mocello
    ** Description: This program prints hello world to the console
    ** Input: none
    ** Output: hello world text
    /*************************************************************************
Data Type

• What is data?
  – Information
  – Ex: `std::cout << “Hello World!” << std::endl;`
  – Simple value
    • Literals, e.g. 23, 79.5, “Hello”, etc.

• What is a data type?
  – Description of the kind of information
    • Primitive Data
    • User Created – (we will cover later)

What are you sending the function?
C++ Primitive Types

- `char`, `double`, `float`, `int`, `long`, `short`, `bool`

- Fundamental
  - `int`: whole numbers, e.g. 45, -89, 0
  - `double`: real numbers, e.g. 2.612, -30.5, 2.3e5
  - `char`: characters, e.g. ‘A’, ‘&’, ‘x’, ‘\’

- Refer to book for types and sizes
Variables

• What is a variable?
  – Memory location with name and type to store value

• What is a declaration?
  – Statement requesting variable w/ name and type
  – Examples:
    double height;
    int age;
Variables/Identifiers

• Identifier: name given to item in program
  – Ex. Variables and Functions
  – Start with letter
    • Letters include: upper-case, lower-case, underscore (_)
    – Followed by sequence of letters and digits
    – Good examples: hiThere, two_plus_two, _hello
    – Bad examples: 5dogs, hi-there, hello there

• Can’t Use Keywords, refer to book...
Variables

• How do we get a value in the variable?
  – Assignment Statement
    int age;
    age = 20;
    Or
    int age = 20;
  – = IS NOT equal to!!!!!
• “gets” or “is assigned”
Printing Variables

- C++: cout
  - Example:
    ```cpp
    std::cout << "The integer value is: " << value;
    ```
  - What about the newline?
    ```cpp
    use endl
    ```
Constants

• What is a constant?

• How do we define a constant?
  – Use of a macro
    • `#define`
    • Placed at top of program
    • No semicolon at end
    • Example: `#define MAX_SIZE 100`

  – Use of `const`
    • Same as declaring variable but const
    • Example: `const int MAX_SIZE = 100;`
Intro to Macros

• C++: <climits>
• Use MIN and MAX macros from library
  http://www.cplusplus.com/reference/clibrary/climits/
  (Note that the values listed are not the values on our system!!)
  – INT_MAX
  – INT_MIN
  – LONG_MAX
  – LONG_MIN
  – SHRT_MAX
  – SHRT_MIN
• Remember unsigned too…
Arithmetic

• Integer Arithmetic
  
  \[
  \text{std::cout} \ll 3/8; \quad /*\text{prints } 0*/ \\
  \text{std::cout} \ll 34/5; \quad /*\text{prints } 6*/ \\
  \]

• Floating Point Arithmetic
  
  \[
  \text{std::cout} \ll 34.0/5.0; \quad /*\text{prints } 6.8*/ \\
  \text{std::cout} \ll 3.0/8; \quad /*\text{prints } .375*/ \\
  \text{std::cout} \ll 3/8.0; \quad /*\text{prints } .375*/ \\
  \]
Type Casting

• Casting
  std::cout << 34 / (int) 5.0; /*prints 6*/
  std::cout << (int) (34 / 5.0); /*prints 6*/
  std::cout << (float) 34 / 5; /*prints 6.8*/
• What is wrong with these?
  std::cout << (int) 34 / 5.0; /*prints 6.8*/
  std::cout << (float) (34/5); /*prints 6.0*/
```cpp
#include <iostream>
#include <climits>
#include <cmath>
using namespace std; //bring in std namespace for cin, cout, endl

int main() {
    unsigned long num = ULONG_MAX;
    cout << num << endl;
    cout << ULONG_MAX << endl;

    //use pow out of the std namespace
    num = (unsigned long)pow(2, sizeof(unsigned long)*8)-1; //minus 1 off end for max

    cout << num << endl;
    return 0; //0 is no error, don't ring houston
}
```