CS 161
Intro to CS I

Beginning to Program
Reminders...

- Read/Start Assignment 1
- Read Chap. 1 & begin Chap. 2
- Exercise Groups, Labs, and Office Hours are happening this week.
- Labs are posted on Canvas.
- Laptop required for Lab.
- Sign-up for exercise group on Exercises page.
- Assignments must compile and run on ENGR!
- Demos start next week (no laptop required).
- Sign-up for demo on home page, after you submit your assignment.
- Don’t be scared!!!!
Odds and Ends

• Lab quizzes just unlock next lab
• Labs are separate from exercises
• Exercises can be done on own or in group session
• Any book will do!!!!
• Help/Make-up for Lab1: Friday (1/7) 3-5pm, KEC 1003
Digital Realm

• Based on discrete #s
  – Specifically: Circuits

• Binary, i.e. base 2
  – 0 or 1

• What base do most people use?
  – What is the range for each digit?

• What is Hexadecimal?, i.e. base 16
  – What is the range for each digit?
Decimal, Binary, & Hex

• Decimal
  – Powers of 10

• Binary
  – Powers of 2

• Base X to Base 10 conversion
  – 32 (base 10): $3 \times 10^1 + 2 \times 10^0 = 32$ (base 10)
  – 100000 (base 2): $1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 32$ (base 10)
  – How do we express 35 (base 10)
    • base 2
    • base 16
More Binary

• What is each digit called?
• What is a Byte?
• How many numbers can be expressed in a Byte?
  – Signed/Unsigned
• What is the smallest number?
• What is the largest number?
Programming

• Writing **code** that a computer can **execute**
  – Does that mean we have to write in binary?

• High-level language
  – Translated Continuously during runtime
    • Interpreted
    • Just in time compilation/caching

  – **Translated Prior/Ahead of time to runtime**
    • High-level -> machine language
    • High-level -> intermediate language
C++ Programming Environment

• Type a program in a .cpp file, vim hello.cpp
• Compile program file, g++ hello.cpp –o hello
• Run the compiled version, hello
• Example: hello.cpp
  
  #include <iostream>
  int main() {
    std::cout << "Hello CS 161 Class!!!";
    return 0;
  }
More C++

• Libraries
  – Example: #include <iostream>
• Functions
  – Perform particular action/computation
  – Requires special function: main
    • int main() {....}
• Statements
  – Ended by semicolon
  – Examples:
    • std::cout << “Hello World”;
    • return 0;
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”;}
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)
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
    ```java
    int age;
    age = 20;
    Or
    int age = 20;
    ```
  – = IS NOT equal to!!!!!
    • “gets” or “is assigned”
Printing Variables

• C++: cout
  – Example:
    std::cout << “The integer value is: ” << value;
  – What about the newline?
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…
Demo...
Reading/Assignments

• Continue Assignment 1
• Finish Chap. 2
• Reminder: Exercise groups, Labs, and Office Hours are happening this week!
• Assignments must compile and run on ENGR!
• Demos start next week (no laptop required).
• Sign-up for demo on home page, after you submit your assignment.