CS 161
Intro to CS I

More Programming, Variables and Constants
Chap. 1.2
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): \(3 \times 10^1 + 2 \times 10^0 = 32 \text{ (base 10)}\)
  – 100000 (base 2): \(1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 32 \text{ (base 10)}\)
  – How do we express 35 (base 10) in base 2 vs. 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?
• Help:
  http://classes.engr.oregonstate.edu/eecs/winter2013/cs161-001/Exam1Review1.txt
Programming

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

• High-level language
  – Interpreted
  – **Compiled**
    • 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**
  ```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 our class style guide
  http://classes.engr.oregonstate.edu/eecs/winter2013/cs161-001/161_style_guideline.pdf
  – 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 p.18 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!\n";`
  – 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 p. 9 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, Appendix 1, p. 915
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:
    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 a variable but preface with const
    • Example: **const int MAX_SIZE = 100;**
Assignment #1 Macros

- C++: `<climits>`
- Use MIN and MAX macros from library
  - INT_MAX
  - INT_MIN
  - LONG_MAX
  - LONG_MIN
  - SHRT_MAX
  - SHRT_MIN
- Remember unsigned too...
Reading/Assignments

• Read Chap. 1.3 - 1.5
Quiz #1

• Get into groups of 3-4
• Discuss Assignment #1.
• Design an algorithm for the solution:
  – What will the variable declarations and assignments look like?
  – How are you going to directly compute the largest and smallest signed short numbers? (Assume a short is 2 bytes.)