CS 161
Intro to CS I
Beginning to Program
Odds and Ends

• Office Hours in KEC 1087 now!
• Assignments must compile and run on ENGR!
• Assignment #1 questions.
Let’s Revisit Binary

• How many numbers can be expressed in a Byte?
  – Signed/Unsigned
• What is the smallest number?
• What is the largest number?
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
 ********************************************************/
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 online resources...
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?
Demo...
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;`
Demo...
Expressions

• What is an expression?
  – Set of operations producing a value
    • Combining simple values
    12 * 4 + 6 * 10 vs. ((12 * 4) + 6) * 10
Expressions cont.

• Pieces of an Expression:
  – Operators
    • Indicate operation, e.g. +, *, /, -, %
  – Operands
    • Values in the expression
  – Evaluation
    • Process of obtaining results from operations on operands
Arithmetic Operators

• Add
  34 + 23
• Subtract
  34 - 23
• Multiply
  2 * 23
• Divide
  40 / 10
• Remainder/Mod
  34 % 5
Arithmetic

• Integer Arithmetic
  std::cout << 3/8;   /*prints 0*/
  std::cout << 34/5;   /*prints 6*/

• Floating Point Arithmetic
  std::cout << 34.0/5.0;   /*prints 6.8*/
  std::cout << 3.0/8;   /*prints .375*/
  std::cout << 3/8.0;   /*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*/
  ```
Precedence

• What is precedence?
  – Binding power of operator
  – (*, /, %) vs. (+, -)

• How do we override precedence?
  – Parenthesis!

• Examples:
  12 * 4 + 6 * 10 vs. ((12 * 4) + 6) * 10
How do we read into a variable in C++?

• Declare a variable
• Read value from user and store at variable location
• How do we do this?

```cpp
#include <iostream>
int main() {
    int x;
    std::cin >> x;
    std::cout << x;
    return 0;
}
```
Demo...