CS 161
Intro to CS I
Variables and Input
Odds and Ends

• Sign up for Assignment 1 Demo
• Peerceptiv Peer Reviews
  – Peer Reviews due Thursday, 11:59pm
  – Back Evaluations due Sunday, 11:59pm
• Assignment 2 posted
• Questions?
Reflections

• Why do we care about climits or sizeof()?

[Handwritten: pow(2,64)]

• Why did we have to typecast pow()?
• Programming Style: please read your class style guide
  – Program Header/Description
  – Placement of {}
  – Indentation: spaces vs. tabs

• String Literal in quotations, “”
  – 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:
  ```cpp
  std::cout << "\"Hello World\"\n";
  ```

- Refer online for common escape sequences:
Data Type

• What is data?
  – Information
  – Ex: `std::cout << “Hello World!” << std::endl;`
  – Literals
    • 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**
  - **short/int/long**: whole numbers, e.g. 45, -89, 0
  - **float/double**: real numbers, e.g. 2.612, -30.5, 2.3e5
  - **char**: characters, e.g. ‘A’, ‘&’, ‘x’, ‘\’

- Signed by default, need to preface with unsigned keyword
  - **unsigned int**
  - **unsigned float**
  - **unsigned char**
Pieces of an Expression

• **Operators**: indicate operation
  – Add +
  – Subtract -
  – Multiply *
  – Divide /
  – Remainder %

• **Operands**: values in the expression

• **Evaluation**: process of obtaining results from operations on operands
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
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*/
  ```
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: 
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/Reading Into Variables

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

- **C++: cin**
  - Example:
    
    ```cpp
    std::cin >> value;
    ```

  easy to remember that `cin is pushing info (>>) into the variable`