CS 161
Intro to CS I

Finish Loops/Begin C++ Strings
Odds and Ends

• Assignment 2 demoed this week!
• Study sessions Tues, Wed, Thur.

5pm 6pm 7pm
How do we read a string of chars?

• User-defined type in string library
  
  ```
  #include <string>
  ```

• Declare/Create type
  
  ```
  std::string mssg;
  ```

• Read with cin or getline
  
  ```
  cin >> mssg; //get a word
  getline(cin, mssg); //get a line of txt
  ```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int i;
    float f;
    string s;

    // ignore all leading whitespace, read characters matching my type
    // until something not of my type or whitespace without moving past
    // what stopped me from reading
    cout << "enter a integer: " << endl;
    cin >> i;
    cout << "i: " << i << endl;

    cout << "enter a float: " << endl;
    cin >> f;
    cout << "f: " << f << endl;

    cout << "enter a string: " << endl;
    cin >> s;
    cout << "s: " << s << endl;
    cout << "length: " << s.length() << endl;
    cout << "first character: " << s.at(0) << endl;

    return 0;
}
Reading multiple types of data...

- What does \texttt{cin} do when reading...
  - Int/Floats
  - Strings

- What does \texttt{getline} do?
```cpp
int main() {
    int i;
    float f;
    string s;

    //ignore all leading whitespace, read characters matching my type
    //until something not of my type or whitespace without moving past
    //what stopped me from reading
    cout << "enter an integer: " << endl;
    cin >> i;
    cout << "i: " << i << endl;

    cout << "enter a float: " << endl;
    cin >> f;
    cout << "f: " << f << endl;

    cout << "enter a string: " << endl;
    //cin >> s;
    getline(cin, s); //read characters until I hit a newline, and move past
    cout << "s: " << s << endl;
    cout << "length: " << s.length() << endl;
}
```
```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {
    int i;
    float f;
    string s;

    cout << "enter a integer: ";
    cin >> i;
    cout << "i: " << i << endl;

    cout << "enter a float: ";
    cin >> f;
    cout << "f: " << f << endl;

    cout << "enter your name: ";
    //cin >> s;
    getline(cin, s); //move past the newline in the buffer leftover from the float
    getline(cin, s); //now read a line
    cout << "s: " << s << endl;
    cout << "length: " << s.length() << endl;
    cout << "first letter in name: " << s.at(0) << endl;

    return 0;
}
```
Extra Credit #2

• Get into groups of 4-5.
• Write your name on one piece of paper.

• How would you determine if user entered a good positive int without using cin.clear() and cin.ignore()?

strings
#include <iostream>

using namespace std;

int main() {
    int i;
    float f;

    cout << "enter an int: " << endl;
    cin >> i;

    // This is a way to check the failbit and reprompt the user if cin fails
    // When will this work and when will it not work?
    while(cin.fail()) {
        cin.clear();
        cin.ignore(256, '\n');
        cout << "enter an int: " << endl;
        cin >> i;
    }

    cout << "enter a float: " << endl;
    cin >> f;

    return 0;
}
```cpp
#include <iostream>
#include <string>
#include <stdlib.h>

using namespace std;

int main() {
    int i;
    float f;
    string s;
    bool error;
    do {
        error = false;
        cout << "Enter an integer: " << endl;
        getline(cin, s);
        for(int i = 0; i < s.size(); i++) {
            if(!(s.at(i) >= '0' && s.at(i) <= '9')) {
                cout << "error!" << endl;
                error = true;
            }
        }
    } while(error);
    i = atoi(s.c_str());
    cout << i << endl;
    cout << "Enter a float: " << endl;
    cin >> f;
    return 0;
}
```
Programming Errors

• Syntax errors
  – Misuse of C++ language
  – How are they caught?

• Logic errors
  – Doesn’t perform task correctly (aka. bugs)
  – How are they caught?

• Runtime errors
  – Stops your program from running
  – How are they caught?
Syntax Error Examples

• Missing main function
• Use of identifier not declared
• Misspelled Words
• Forget a Semicolon
• Forget Required Keyword
• Missing quote, curly brace, and parenthesis
• Use of single quotes instead of double
Logic Error Examples

- Poorly written programs
  - Add instead of subtract (incorrect operation)
  - Using last two digits for date
  - Same error message for different errors
  - Program that never ends
  - Add one to the largest integer (could be syntax)
Runtime Error Examples

• Segmentation fault or Core dump
  – Read a file that doesn’t exist
  – Go outside of memory bounds
  – Run out of memory
  – Divide by variable that is zero
Debugging Errors

• Syntax:
  – **READ compiler errors** (pay attention to line #)
  – Use **google** to search for error

• Logic/Runtime
  – Use **std::cout** to find where the code is breaking
    • **Print variable values**
    • **Print indicator messages**
  – **Trace** through the code
  – **Comment** out code