CS 161
Intro to CS I

Finish Loops/Begin C++ Strings
Odds and Ends

• Assignment 2 demoed this week!
• Study sessions Tues, Wed, Thur.
How do we read a string of chars?

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

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

• Read with cin or getline
  
  ```
  cin >> mssg;  //get a word
  getline(cin,mssg);  //get a line of txt
  ```
Demo...
Reading multiple types of data...

• What does cin do when reading...
  – Int/Floats
  – Strings

• What does getline do?
Demo...
```cpp
#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;
}
```
Solution...
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