CS 161
Intro to CS I

Finish Loops/Begin C++ Strings
Odds and Ends

• Peer Reviews on design due Thursday.
• Exam I next Wednesday, 2/6/2019
```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;
}  
```
```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
Decomposition

• Divide Problem (task) Into Subtasks
  – Procedural Decomposition
  – Examples: cooking, cleaning, etc.

• Incremental Programming
  – Iterative Enhancement (Stepwise Refinement)

• Examples: Replicating Code
Functions

• What is a function?
  – Block of code to perform action/subroutine

• When have we seen functions already?
  – Predefined

• What is the purpose?
  – Reduce
  – Reuse
  – Readability
Predefined Functions

- `sqrt()`
- `pow()`
- `abs()`
- `rand()`
- `srand()`
- What is the difference b/w `srand()` and others?
Procedural Decomposition

• Functions
  – int **main**() {
  
  – User defined
    void draw_box() {
    }

• Function Call
  – draw_box();
#include <iostream>
using std::cout;

void draw_box();  //Declare function

int main() {
    cout << “+--------+
”;  
    cout << “|           |
”;  
    cout << “+--------+
”;  
    cout << “+--------+
”;  
    cout << “|           |
”;  
    cout << “+--------+
”;  
    return 0;
}

#include <iostream>
using std::cout;

void draw_box();  //Declare function

int main() {
    draw_box();  //Use function
    return 0;
}

void draw_box() {  //Define function
    cout << “+--------+
”;
    cout << “|           |
”;  
    cout << “+--------+
”;  
    cout << “+--------+
”;  
    cout << “|           |
”;  
    cout << “+--------+
”;  
    return 0;
}
#include <iostream>

void draw_box();
void draw_top_bottom();
void draw_sides();

int main()
{
    draw_box();
    return 0;
}

void draw_box()
{
    draw_top_bottom();
    draw_sides();
    draw_top_bottom();
}

void draw_top_bottom()
{
    std::cout << "+--------+
    
}

void draw_sides()
{
    std::cout << "|           |
    
}
Generalization

• Does a function make a task more specific or more general?
  – Justification
  – Examples
void Functions

- Doesn’t return a value
- Still has arguments/parameters
Programming Demo