CS 161
Intro to CS I

Decomposition/Functions
Chap. 3
Finish Debugging

- Pay attention to compiler messages
- Use google to search for errors
- Print indicator messages
- Print variable values
- Trace through the code
Finish Error Handling

• Prevent Divide/Mod by Zero

```cpp
#include <iostream>
using std::cout;

int main() {
    int x, y;
    cout << "Enter the numerator: ";
    cin >> x;
    cout << "Enter the denominator: ";
    cin >> y;
    while (y == 0) {
        cout << "Denominator can't be zero, enter new input: ";
        cin >> y;
    }
    cout << "X divided by Y is: " << x / y << "\n";
    return 0;
}
```

Decomposition

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

• Incremental Programming
  – Iterative Enhancement (Stepwise Refinement)

• Examples: Replicating Code
Procedural Decomposition

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

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

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

#include <iostream>
using std::cout;

void draw_box();  //Declare function

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

void draw_box() {  //Define function
cout << "+--------+
    cout << "|           |
    cout << "+--------+
    return 0;
}
Functions Calling Other Functions

```cpp
#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 << "|           |
    
}
```

OSU Oregon State University
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
Generalization

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