CS 161
Intro to CS I

Decomposition
Chap. 3.1-3.3
```cpp
#include <iostream>
#include <string>
#include <cstdlib>

using namespace std;

int main() {
    char cont;
    string shape, str_num;
    int x;

    cout << "Do you want to continue (Y or N):"; 
    cin >> cont;

    cout << "Please enter a number: ";
    cin >> str_num;

    // Take a C++ string and change to C string and then check to see if int
    while(! (x=atoi(str_num.c_str()))) { 
        cout << "Again, please enter a NUMBER: "; 
        cin >> str_num;
    }

    cout << "You entered: " << x << endl;

    //while(cont=='Y' || cont=='y') {
    //    cout << "we want to cont, enter shape: ";
    //    cin >> shape;
    //    cout << "Do you want to continue (Y or N):";
    //    cin >> cont;
    //}
    */

    for( ; cont=='Y' || cont=='y'; cin >> cont) {
        cout << "we want to cont, enter shape: ";
        cin >> shape;
        cout << "Do you want to continue (Y or N):";
    }

    return 0;
}
```
Handling Errors – Quiz #3

• Get into groups of 4-5.

• What can we do to prevent these errors?
  – Overflow
  – Divide by zero

• Discuss Assignment #3, design a solution, and provide your algorithm using pseudocode.
  • Where do you need a loop (or loops), and what kind of loop(s) are you going to use?
Decomposition

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

• Incremental Programming
  – Iterative Enhancement (Stepwise Refinement)

• Examples: Assignment #2
Procedural Decomposition

• Functions
  – int main() {
  
  – User defined
  
    void draw_box() {
  
• Function Call
  – draw_box();
Procedural Decomposition

```c
#include <stdio.h>
int main() {
    printf("+---------+\n");
    printf("|       |\n");
    printf("+---------+\n");
    printf("---------+\n");
    printf("|       |\n");
    printf("---------+\n");
    return 0;
}

#include <stdio.h>
void draw_box(); //Declare function
int main() {
    draw_box(); //Use function
draw_box();
    return 0;
}
void draw_box() { //Define function
    printf("+---------+\n");
    printf("|       |\n");
    printf("+---------+\n");
}
```
Procedural Decomposition

#include <stdio.h>
int main() {
    printf(“+--------+\n”);
    printf(“|       |\n”);
    printf(“+--------+\n”);
    printf(“+--------+\n”);
    printf(“|       |\n”);
    printf(“+--------+\n”);
    return 0;
}

#include <stdio.h>
void draw_box() { //Define function
    printf(“+--------+\n”);
    printf(“|       |\n”);
    printf(“+--------+\n”);
    printf(“+--------+\n”);
    printf(“|       |\n”);
    printf(“+--------+\n”);
}

int main() {
    draw_box(); //Use function
draw_box();
    return 0;
}
#include <stdio.h>

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

main() {
    draw_box();
}

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

void draw_top_bottom() {
    printf("+--------+
");
}

void draw_sides() {
    printf("|
");
}
Scope

- Part of program in which a declaration is valid
- Local variable
  - Declared inside a function only accessible inside function
- Localizing variables
  - Declaring variable in innermost scope
Illegal access outside loops

for(x = 0; x < 10; x++) {
    int y = 10;
    printf(“The value of x * y is: %d\n”, x*y);
}
printf(“The value of y is: %d\n”, y); /*y outside scope*/

• How do we fix this?
• What about if/else blocks?
Illegal access in functions

```c
int main () {
    int x=2, y=3;
    compute_sum();
    sum = x+y;  //error: sum hasn’t been declared
    return 0;
}
void compute_sum() {
    int sum = x+y;  //error: x and y outside scope
}
```
Functions

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

• When have we seen functions already?
  – Decomposition

• What is the purpose?
  – Reduce
  – Reuse
  – Readability
Generalization

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