CS 161
Intro to CS I

Loops/Begin Functions
Odds and Ends...

• Test Friday
• Study Sessions: Wed 6-7:30 and Thur 5-7:30
• Last week to demo Assignment 2!!

• Read: conditionals (if/else and switch), loops (for, while, do-while), and begin functions
Demo...

```cpp
bool bad_input;
// cin ignores all leading whitespace,
// then reads characters of its expected
// type until it gets to a character not
// of its type OR whitespace without
// consuming the char not of its type or whitespace
do {
    bad_input=false; // assume good input each time, until given otherwise
    // can use to see if a cin failed and then clear the failbit and ignore bad
    // *if(cin.fail()){*
    cin.clear(); // clear the failbit
    cin.ignore(256, '"'); // ignore/disregard our the input
    /*}
    cout << "enter an int: ";
    cin >> s; // what happens when you enter a non-integer or newline/enter?
    cout << "value of x: " << s << endl;
    for(int i=0; i<s.size(); i++){ // for all the chars in s
        if(!(s.at(i)=='0' || s.at(i)=='9')) // check each to see if 0-9 digit
            bad_input=true;
    }
} while(bad_input);

x=stoi(s.c_str()); // now you know the string is good, so change ascii to int
```

--- INSERT ---
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();
Procedural Decomposition

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

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

void draw_box();  //Declare function

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

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

#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 << “+--------+\n”;
}

void draw_sides() {
    std::cout << “|           |\n”;
}
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
Predefined Functions

- sqrt()
- pow()
- abs()
- rand()
- srand()
- What is the difference b/w srand() and others?
void Functions

• Doesn’t return a value
• Still has arguments/parameters

• Can we write a \texttt{void good_pos_int()}?
• Is it more useful to return a value?
Demo...
Scope (Visibility)

• 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

```cpp
for(x = 0; x < 10; x++) {
    int y = 10;
    cout << “The value of x * y is: ” << x*y << endl;
}
cout << “The value of y is: ” << y << endl; /*y outside scope*/
```

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

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
}
More about break, exit, and return

- **break** – used with switch and loops, breaking out of the closest associated case or loop (for, while, or do while). **This statement can only occur in a loop or case**, otherwise the compiler yells!

- **return** – leave the current function, which exits the program when in the main() function. You can put this **anywhere inside any function**, otherwise the compiler yells!

- **exit()** – exit the entire program, no matter where this is encountered. You can put this **anywhere inside any function**, as long as you include `<cstdlib>`, otherwise the compiler yells!
Group Exercise #3

• Write your own pow...