ECE/CS 151
Intro to Programming I

Decomposition and Scope
Chap. 4.3-4.4
Odds and Ends...

• Revisiting scanf options
  – %d and %c do not get ‘\0’ (NULL)
  – %s does get ‘\0’
  – What does this mean for %1s?
  – What if user supplies char while reading int?

• Assignment #3
  – You do not need to know how to integrate
  – How would you check you answer?
    • (evaluate 2/3*X^3 at b) – (evaluate 2/3*X^3 at a)
Programming Demo

```c
#include <stdio.h>

int main () {
    int q=-1;
    char c;

    printf("Enter a number: ");
    scanf("%d", &q);
    if(q==-1) {
        printf("Enter a number dummy\n");
        scanf("%s");
        printf("Enter a number: ");
        scanf("%d", &q);
    }
    /*getchar(); Read the newline character*/
    printf("Enter a character: ");
    scanf("%c%c", &c);
    printf("%d %c\n", q, c);
    return 0;
}
```

"scanf_ex.c" 21L, 402C written 18,4 All
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;
}

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");
}
```

Functions Calling Other Functions

#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("+-------+\n");
}

void draw_sides()
{
    printf("|       |\n");
}
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
Quiz #4

- Get into groups of 3-4, or you can choose to work alone. Discuss Assignment #3, design a solution, and provide your algorithm using pseudocode.
  - Where do you need a loop, and what kind of loop are you going to use?
  - How are you going to use decomposition?