CS 161
Intro to CS I

RECAP Chap. 1 – 4.1
Recursive Functions Chap. 13

Exam 1 Results

- Average: 74
- As: 18
- Bs: 10
- Cs: 18
- Ds: 28
- Fs: 14

- What went wrong?

Lab #6...

- Understand the right answers
- Look into recursion
Recursion

• What is it?
  – Function that calls itself 1 or more times
  – Has a base case for stopping

Example: Drawing Rectangles

• Iterative Solution:
  ```cpp
  void draw_rect(int i) {
    for( ; i>0; i--){
      cout << "******n";
      cout << "*   *n"
      cout << "******n"
    }
  }
  ```

• Recursive Solution
  ```cpp
  void draw_rect(int i) {
    if(i>0) {   //Base case
      draw_rect(--i);  //Recursive call
      cout << "******n";
      cout << "*   *n"
      cout << "******n"
    }
  }
  ```
Example: Factorial

- Definition
  0! = 1;
  n! = n * (n-1) * ... * 2 * 1 = n * (n-1)! ; n > 0

Iterative Factorial

factorial(0) = 1;
factorial(n) = n*n-1*n-2*...*2*1;

```c
long factorial(int n) {
    long fact;
    if(n==0)
        fact=1;
    else
        for(fact=n; n > 1; n--)
            fact=fact*(n-1);
    return fact;
}
```

Recursive Factorial

factorial(0) = 1;
factorial(n) = n*factorial(n-1);

```c
long factorial(int n) {
    if (n == 0)  // Base case
        return 1;
    else
        return n * factorial(n - 1);  // Recursive call
}
```