

# CSCI-1200 Computer Science II — Fall 2006

## Lab 8 — Recursion I

### Checkpoint 1

Here is a silly pair of mutually-recursive functions:

```
bool Odd(int number);

bool Even(int number) {
    if (!Odd(number)) return true;
    else return false;
}

bool Odd(int number) {
    if (!Even(number)) return true;
    else return false;
}
```

Type this code into your programming environment and try it out. Use the debugger to see what goes wrong. Rewrite the code, but you are only allowed to use subtraction and testing if something is equal to zero. Also `Odd` cannot directly call itself and `Even` cannot directly call itself.

**To complete this checkpoint:** Show a TA what goes wrong in the original version and how you fixed the problem.

### Checkpoint 2

Study the recursive definition below which computes the  $n$ th Fibonacci number. Draw the *complete activation record hierarchy* that results from the function call `fib_a(4)`. This recursion example is complex to analyze exactly, but find a reasonable *upper bound* on the number of calls to `fib_a` as a function of  $n$ .

```
int fib_a(int n) {
    assert (n >= 0);
    if (n == 0) return 1;
    if (n == 1) return 1;
    return fib_a(n-1) + fib_a(n-2);
}
```

**To complete this checkpoint:** Show a TA your diagram and analysis.