

# Global variables

- Suppose we wanted to count the number of times fib calls itself recursively
- Can do this using a global variable
- So far, all functions communicate with their environment through their parameters and return values
- But, (though a bit dangerous), can declare a variable to be global – means name is defined at the outermost scope of the program, rather than scope of function in which appears

# Example

```
def fibMetered(x):  
    global numCalls  
    numCalls += 1  
    if x == 0 or x == 1:  
        return 1  
    else:  
        return fibMetered(x-1) + fibMetered(x-2)  
  
def testFib(n):  
    for i in range(n+1):  
        global numCalls  
        numCalls = 0  
        print('fib of ' + str(i) + ' = ' + str(fibMetered(i)))  
        print('fib called ' + str(numCalls) + ' times')
```

# Global variables

- Use with care!!
- Destroy locality of code
- Since can be modified or read in a wide range of places, can be easy to break locality and introduce bugs!!