

## More Functions with Return Values

CS303E: Elements of Computers and Programming  
July 16, 2012

## Reminder: Variable Scope and Functions

- The *scope* of a variable is the part of the program in which that variable is accessible
- A function parameter's scope is its functions

## Pass By Value

- A function cannot change the value of the argument that is passed to it
  - The value of the argument in the calling statement does not change, no matter how the function manipulates the parameter
- Some programming languages do this, others use *pass by reference*
  - It is important to find out which!

## Pass By Value: Example

```
def main():
    arg1=35
    print "arg1 is", arg1
    changeIt(arg1)
    print "after call," + \
        "arg1 is", arg1

def changeIt(par):
    #recall that initially
    #par is set to the
    #value of arg1
    par=-1
    print "par is", par
```

What happens when this program executes?

### iClicker Question

What does *pass by value* mean?

- A. Changes to a function's argument are reflected in the calling function
- B. Changes to a function's argument are NOT reflected in the calling function

### Returning Multiple Values

- A function may return a *tuple* which is a collection of values.

– So... it's not *really* multiple values, but one value that is itself a collection of values

Example:

```
return a,b
```

But how do you assign multiple values?

### Simultaneous Assignment

- Use *simultaneous assignment* to assign multiple values in one statement

```
a,b=2,3
print "a=%d b=%d" % (a,b)
```

Output:  
a=2 b=3

### Returning Multiple Values: Example

```
def divide(a,b)
    #return dividend and remainder in a tuple
    return a/b, a%b

def main():
    div,rem=divide(13,8)
    print "%d/%d=%d R %d" % (13,8,div,rem)
```

iClicker Question	
<p>■ What are the values of a and b?</p> <pre>def add_sub(x,y):     return x+y,x-y def main():     a=4     b=5     a,b=add_sub(a,b)</pre>	<p>A. a=4 b=5 B. a=9 b=9 C. a=9 b=-1 D. a=-1 b=9</p>

Exercise	
<p>Write a function minMax() that takes 2 arguments, a and b, and returns a tuple containing the minimum and maximum of a and b, in that order. Then write a main function that calls minMax() on all combinations of inputs a and b such that <math>1 \leq a \leq 5</math>, <math>0 \leq b \leq 3</math>.</p>	

Group Exercise	
<p>Write a program to simulate a digital clock. Your program should print the values of each second like this:</p> <pre>00:00:00 00:00:01 00:00:02 ... 23:59:59</pre>	
<p>You will need the following code:</p> <pre>import time  time.sleep(1) #tells #program to wait for 1 #second before continuing</pre>	