

## CSI31 Lecture 17

### **Topics:** *Chapter 6. Defining Functions*

6.5.2 Functions that modify parameters

6.6 Functions and program structure (covered in Lecture 16)

### **Addition to HW#13:**

add at least two tests to your program  
(similar to the way it is done in tests.py)

## Functions may return more than one value.

### Example:

the following program has a function that takes two parameters and returns their sum and their product.

```
def main():
    x,y=input('Please, input two decimal numbers:')
    sum_,prod=sum_prod(x,y)

    print 'The sum of numbers %0.3f %0.3f is %0.3f,
and their product is %0.3f' % (x,y,sum,prod)

def sum_prod(x,y):
    return (x+y), (x*y)

main()
```

## Can functions modify the parameters?

«Python passes all parameters by value»

- («original» values cannot be changed inside a function)

see example: [prog1.py](#)

Other languages have a mechanism that allows to change parameter's values, that is called «call by reference». Python doesn't have it.

So we will have to **return** modified value. See [prog2.py](#)

**!** Functions may modify variables that are mutable objects (lists, graphics objects)

**Mutable** means that the value of an item in a list can be modified with an assignment statement.

(! recall that strings are not mutable)

See [prog3.py](#)

## Testing and debugging

Testing is an essential part of a program developing process.  
A program cannot be released unless it passes some testing.

Here is an example of how to write small test functions: see [tests.py](#)

An error in a program is called a [bug](#).

The process of finding and eliminating the bugs is called [debugging](#).

I showed how to use Active State Python Debugger in class