## Objectives

- More on functions
  - Passing parameters
- Top-down design
- Files
  - Writing files
  - Handling numeric data



## Review

- What is the keyword we use to create a new function?
- How do we get output from a function?
- What happens in the program execution when a function reaches a `return` statement?
- Why do we write functions?
- What makes a good function?
- How should you comment your functions?
- What is the name for the process for changing a program to improve readability/organization/ readability without changing functionality?



Immutable vs Mutable Parameters

## PASSING PARAMETERS



## Passing Parameters

- Only ***copies*** of the actual parameters are given to the function
  - For **immutable** data types Which are?
- The *actual* parameters in the calling code do not change
- **Swap example:**
  - Swap two values in script
  - Then, put into a function




## Immutable Data is Passed by Value

```
def swap(a, b):
    tmp = a
    a = b
    b = tmp
    print(a, b)

x = 5
y = 7

swap(x, y)

# at the end, y should be 5 and x should be 7
print("x =", x)
print("y =", y)
```



## Immutable Data is Passed by Value

```
def swap(a, b):
    tmp = a
    a = b
    b = tmp
    print(a, b)

x = 5
y = 7

swap(x, y)

# at the end, y should be 5 and x should be 7
print("x =", x)
print("y =", y)
```

This code does not have the desired effect.

Because the integers are passed *by value*, the value of x and y are not changed by the call to the swap function.

## Lists as Parameters to Functions

If a list that is passed as a parameter into a function is **modified *in* the function**, the list **is modified *outside* the function**

- Lists are **not** passed-by-value/copied
- Different from immutable types (e.g., numbers, strings)

- Parameter is actually a **pointer** to the list in memory



## Problem: Sort a list of 3 numbers, in descending order

```
# order list such that list3[0] >= list3[1] >= list3[2]
def descendSort3Nums( list3 ):
```

Called as:

```
list = …
descendSort3Nums(list)
print(list)
```

How implemented with list methods?
Can we do this using only 3 comparisons?



## Descend Sort a List w/ 3 elements

```
def descendSort3Nums(list3):
    if list3[1] > list3[0]:
        # swap 'em
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp

    if list3[2] > list3[1]:
        tmp = list3[1]
        list3[1] = list3[2]
        list3[2] = tmp

    if list3[1] > list3[0]:
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp
```

```
def main():
    list = [1,2,3]
    descendSort3Nums(list)
    print(list)
```

Function does **not** *return* anything. Simply modifies the `list3` parameter.



## Review: Files

- How do we create `file` objects?
- How do we read from files?
- What should we do after opening a file?



## Review: Common File Methods

| Method Name | Functionality |
|---|---|
| `read()` | Read the entire content from the file, returned as a string object |
| `readline()` | Read one line from file, returned as a string object (which includes the "\n"). If it returns "", then you've reached the end of the file |
| `write(string)` | Write a string to the file |
| `close()` | Close the file. Must close the file after done reading from/writing to a file |



## Writing to a File

- Create a file object in **write** mode:
  - `myFile = open("output.txt", "w")`

- Example: create a file from user input
  - `file_write.py`

What happens if you execute the program again with different user input?

## Handling Numeric Data

- We have been dealing with reading and writing *strings* so far
  - Read from a file: get a string
  - Write to file: use a string
- What do we need to do to **read numbers** from a file?
- How can we **write numbers** to a file?



## Handling Numeric Data

- We have been dealing with reading and writing *strings* so far
  - Read from a file: get a string
  - Write to file: use a string
- What do we need to do to **read numbers** from a file?
  - Cast as a numeric type, e.g., `int` or `float`
- How can we **write numbers** to a file?
  - Cast number as a `str`



## Problem: Temperature Data

- **Given**: data file that contains the daily high temperatures for last year at one location
  - Data file contains one temperature per line
  - Example: `data/florida.dat`
- **Problem**: What is the average high temperature (to 2 decimal places) for the location?

**Rule of Thumb**: Always look at data file before processing it



## Looking Ahead

- Lab 7 due Friday
- Broader Issue due Friday