**CSC/CPE 202: Spring 2020**

**LAB-1 (Due 4/13)**

**Working with Linux environment in the Lab:**

1. For accessing university Linux servers first follow the given link: https://calpoly.atlassian.net/wiki/spaces/CPKB/pages/2425047/Set+up+VPN

2. **For Windows users:**

Download the SSH (Secure Shell Client). Use one of the servers: unix1, unix2, unix3, or unix4 to connect to server.

host name: unix1.csc.calpoly.edu

username: your CalPoly username (just username)

3. **For Mac users:**

Open terminal in your Mac:

```
ssh CalPoly-Username@unix1.csc.calpoly.edu
```

or you can use unix2, unix3, or unix4.

## Part 1: Review

In the plantes.py file, add code to implement the functionality shown by the following sample run.

**Sample Run:**

```
What do you weigh on earth? 136

On Mars you would weigh 51.68 pounds.

On Jupiter you would weigh 318.24 pounds.
```

### Important Information and Requirements:

- To calculate a person's weight on Mars, multiple their weight on earth by 0.38.
- To calculate a person's weight on Jupiter, multiple their weight on earth by 2.34.
- Your output must match my sample output exactly. Make sure there is a blank line in between prompting for the user's weight and displaying the results.
- Your program may only use the print function once and format output using format function.
- Run the given unit tests in planets_tests.py to check your solution!

## Part 2: Recursive

Experienced programmers know how important it is to understand the problem and develop a solution before writing any code. Why? Because this saves time and effort in the long run. Drawing pictures and writing pseudo code clarify both the problem and how to solve it. If you cannot solve simple versions of the problem by hand using your algorithm, how can you expect to write a program that will solve the problem.

When writing pseudocode, at first do **not** give a complete program. Give just enough detail that your intentions are clear and unambiguous, but do not provide extraneous details (complex syntax and/or variable declarations) which are **otherwise obvious**. Thinking through examples is especially important. This saves much time later in reducing rework, finding problems, etc.

**Note: For each function write pseudocode, signature, and purpose statement.**

## Recursion and Python practice

In Lab1.py you need to add following functions:

- Write an iterative function to find the maximum in a list of integers. (do not use library function like max or sort.) The recursive solution is given. Your job is converting recursive implementation to iterative one.
- Write a recursive function to reverse a string (do not use reverse library function).
- Write a recursive function to search a list of integers using binary search along with test cases. The function returns its index if the **target** of the search is in the list else return **None**. If the list is empty, it returns **None**. You had the iterative implementation of Binary Search in CPE101.

In Lab1_test_cases.py you need to add three test cases for each function.

Demo your work before deadline ends.

**Submissions:**

1) planets.py
2) planets_tests.py
3) Lab1.py
4) Lab1_test_cases.py

# For folks new to Python, or Python unittest, or who need review

- the unittest framework. I recommend watching the first 20 minutes or so of the following video if you are unfamiliar with testing in Python. https://www.youtube.com/watch?v=6tNS--WetLI

- The basic Python Tutorial
  https://docs.python.org/3/tutorial/

- This site gives the essentials.

Python for Java Programmers: http://python4java.necaiseweb.org/Fundamentals/Fundamentals

## Videos on specific topics you will need to know.

**Installing Python on Mac/Windows**: https://www.youtube.com/watch?v=YYXdXT2l-Gg&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU

**Strings**: https://www.youtube.com/watch?v=k9TUPpGqYTo&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=2

**Lists, etc. (Lists are what you need for now)** https://www.youtube.com/watch?v=W8KRzm-HUcc&index=4&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU

**Conditionals:** https://www.youtube.com/watch?v=DZwmZ8Usvnk&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=6

**Loops:** https://www.youtube.com/watch?v=6iF8Xb7Z3wQ&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=7

**Functions**: https://www.youtube.com/watch?v=9Os0o3wzS_I&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=8

**Modules:** https://www.youtube.com/watch?v=CqvZ3vGoGs0&list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU&index=9