CS 30 Lab 2 — Introduction to Python and Emacs

This lab gives you practice writing and running simple Python programs using Emacs. You are free to work with a partner if you like, and to talk things over with your lab-mates. Don't hesitate to call me or Dan over for help or answers to questions.

---

1. Log into your Linux account, right-click on your Desktop to open a terminal window, and then start Emacs by typing **emacs &** at the Linux prompt (or, alternatively, by choosing Applications->Accessories->Emacs21 from the menus at the bottom left of the screen). Once in Emacs, type **Control-h t** to enter the tutorial. Spend a few minutes going through the first part of the tutorial, to become familiar with the basic editing commands.

2. Next, open a new file (by typing **Control-x Control-f**) and call it **paragraph.txt**. The **.txt** extension will automatically cause Emacs to enter Text mode. Write a paragraph or two telling me a little about your background. Remember to save the file when you're done, using **Control-x Control-s**. I'm interested in knowing the following:

   - How much previous computer experience do you have?
   - Are you comfortable with Windows, Macintosh, and/or Linux?
   - Have you ever taken a CS course or studied programming before?
   - Why are you taking CS 30 and is there anything in particular you hope to learn?
   - Anything else you'd like me to know about you?

3. Return to the Linux command line and type **/common/cs/submit/cs30-submit**. This will run the homework handin program that we'll be using this semester. Turn in your **paragraph.txt** file when prompted.

4. Using Emacs, type in the **chaos.py** program from Chapter 1 of the textbook (which we did in class this morning), and save it in a file of the same name. Then start the Python interpreter in a terminal window and try out the program:

```
>>> import chaos
>>> chaos.main()
```

5. Modify **chaos.py** so that it prints out 20 values instead of 10. Save your changes in Emacs, and then type the following command at the Python prompt:

```
>>> reload(chaos)
```

   This will reload your program into Python. Test the new version to make sure it works. Next, modify it so that the number of values to print is determined by the user. Make sure to test this version, too. You will have to add a line near the top of the program to get another value from the user:

```
n = input("How many numbers should I print? ")
```

6. Assuming that 1 minute of time is worth $25 (surely a gross underestimate!), write a program to convert time into money. Your program should ask the user for the number of minutes to convert, and then print out a message giving the equivalent dollar amount. Be sure to test your program on several different input values.

(continued on back)

7. Do problem 1 (volume and surface of a sphere) on page 72 of the textbook.

8. Do problem 2 (cost of a pizza) on page 72 of the textbook.

9. Do problem 9 (area of a triangle) on page 73 of the textbook.

10. Write a program to find the sum of the first *n* natural numbers, where the value of *n* is provided by the user. Hint: use a for-loop.

11. Write a program to find the sum of the *cubes* of the first *n* natural numbers, where the value of *n* is provided by the user. Do this by modifying your program from problem 7.

12. Write a program to sum a series of numbers entered by the user. The program should first prompt the user for how many numbers are to be summed. It should then input each of the numbers and print a total sum. Hint: Use an extra variable called `total`, initialized to 0, to keep track of the running sum. The structure of your program will be similar to the previous ones.

13. Write a program that finds the average of a series of numbers entered by the user. As in the previous problem, the program will first ask the user how many numbers there are. Note: the average should always be a floating-point value, even if the user inputs all integer values.

Slightly More Challenging

1. Do problem 15 (approximating π) on page 74 of the textbook.

2. Do problem 16 (Fibonacci numbers) on page 74 of the textbook.

3. Do problem 17 (Newton's method) on page 74 of the textbook.