UNIVERSITY OF MICHIGAN ❧ SCHOOL OF INFORMATION

Assignment 3 – Writing some Python

Due Date: Friday January 30, 2008 at 5:00PM

The purpose of this exercise is to get you writing some small Python programs. You will write three programs for this assignment:

- A program to convert Celsius to Fahrenheit
- A program to prompt for a file name and count the lines in the file
- A program to display the lines in the file converted to upper case

These programs come from Chapters 2 and 4 of the Zelle book and the associated lectures as well.

General Guidance

These three programs are pretty simple – you should be able to construct them reading the book and looking through the slides for Chapters 2 and 4.

One pattern we see when beginning students are trying to write a simple program is that they get part way done and get stuck on a little problem – like a bad indentation error or something – when you can't fix the problem right away (often the error messages are a bit cryptic) – you conclude that everything you know is wrong and throw the whole program away and try to rewrite it from some other approach.

Often this second approach is less "correct" than the first approach – and has lower chance of success than the first approach. So after an hour of pain, the second try fails and you make a fresh third try – which is an even weirder approach – resulting in another hour of pain.

By this point – you completely have lost what you learned – and often the 10-line program with one small error has become a 40-50 line monster with lots of bits of code typed in randomly from the slides, books, and Google. You look at the monster code and decide that programming is impossible. (Actually it is impossible to just paste in random bits and get it to work).

So the right reaction when you kind of have your program looking like you think it should look – but there is one silly error – **get help**. If you get help and quickly get past the little error – you will realize that you actually knew 95% of the program – and yeah there is some little trick about mixing tabs and spaces and you just won't do that again.

It takes a few minutes to get some help mailing your GSI or mailing the whole class list in a pinch. Don't just get mad and try to put random things in your code to make it work.

First Program: Fahrenheit to Celsius

This program is very similar to the example on page 28 of the Zelle book – it is just the opposite conversion – you enter a Fahrenheit number and the program prints out Celsius. Here is an example of the program running:

```
python faren.py
What is the Fahrenheit temperature? 212
The temperature is 100.0 degrees Celsius.

python faren.py
What is the Fahrenheit temperature? 72
The temperature is 22.2222222222 degrees Celsius.
```

The next assignments require a data file.

The faren.py should be around 10 lines of Python code.

Getting the File For the Data

Grab a file from CTools under **Assignments -> Data** – for this assignment you will only use the file named **mbox-short.txt**

Create a folder for this assignment to hold both the data file and the Python programs that you write. Go to CTools and click on **mbox-short.txt** and then use **File -> Save As** in your browser to put the file into the right folder on your computer.

If you look at the text file in an editor such as JEdit it is just lists of mail messages concatenated together that look as follows:

```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Received: from paploo.uhi.ac.uk (localhost [127.0.0.1])
        by paploo.uhi.ac.uk (Postfix) with ESMTP id 5F919BC2F2;
Message-ID: <200801051412.m05ECIaH010327@nakamura.uits.iupui.edu>
Date: Sat, 5 Jan 2008 09:12:18 -0500
To: source@collab.sakaiproject.org
From: stephen.marquard@uct.ac.zaSubject: [sakai] svn commit: r39772 - content/branches

Details: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772

Author: stephen.marquard@uct.ac.za
Date: 2008-01-05 09:12:07 -0500 (Sat, 05 Jan 2008)
New Revision: 39772

From louis@media.berkeley.edu Fri Jan  4 18:10:48 2008
Return-Path: <postmaster@collab.sakaiproject.org>
```

Second Program: Counting Lines in the File

This program will prompt for a file name and open that file – it will read all of the lines in the file and print out the name of the file and the count of the lines in the file as shown below:

```
python count.py
```

```
Enter a file name: mbox-short.txt
mbox-short.txt contains 1910 lines
```

Take a look at Page 107-111 for detail on how to deal with files.

The count.py should be around 10 lines of Python code.

Third Program: SHOUTING THE CONTENTS OF A FILE

This program will read through a file and print the contents of the file (line by line) all in upper case. Executing the program will look as follows:

```
python shout.py
Enter a file name: mbox-short.txt
FROM STEPHEN.MARQUARD@UCT.AC.ZA SAT JAN  5 09:14:16 2008
RETURN-PATH: <POSTMASTER@COLLAB.SAKAIPROJECT.ORG>
RECEIVED: FROM MURDER (MAIL.UMICH.EDU [141.211.14.90])
      BY FRANKENSTEIN.MAIL.UMICH.EDU (CYRUS V2.3.8) WITH LMTPA;
      SAT, 05 JAN 2008 09:14:16 -0500
```

Start by printing out the lines in the file without converting them to upper case. If you end up with an extra blank line – you are getting a “newline” with your input. The **rstrip()** function from the string library is a good way to get rid of this trailing whitespace.

```
    line = string.rstrip(line)
```

Of course you should use the **upper()** function from the string library to do the capatialization.

If you look through the Zelle book –there are other ways to read a file that may work as well – there is more than one way to solve this problem – all are OK.

The shout.py should be around 10 lines of Python code.

What to Hand In

Hand in three screen shots of the command line program – one for each of your programs successfully running. And also hand in the three Python source code files.

So your assignment should upload six files total.

Remember to hand in JPG for Mac systems and PNG for Windows systems for your screen shots.