CSC 111aD - Fall 2013 - Project_3:
Due 11/7/13,12pm (noon)

Instructions

- Submit to Sakai on or before the due date. If not complete, submit what you have.
- Program files should use the following file naming format:
  - Lastname_Firstinitial_Project_Project#.py
  - For example, for this assignment, the project is Project #3, the file would be named: *Thomas_S_Project_3.py*
- All programming assignments must follow the style guide, include helpful and appropriate comments, and have meaningful variable names.

Write a Python program to summarize the sales at a restaurant for a single day. The idea is that sales are recorded by servers in a simple text data file during the day. At the close of the day your program will read the text file and summarize the sales for the day, as instructed below. The remainder of these instructions will refer to the data file as the *receipts* file.

Each data file will have a 21 character name matching the following pattern: receipts-mm-dd-yy.txt for example, receipts-10-30-13.txt. Several data files can be downloaded from Sakai->Resources.

Each line of a *receipt* file will have a fixed format consisting of five fields separated by a single ":". The fields are as follows:

Field 1 – contains either the word "cash" or a credit card number. The credit card number may or may not be a valid number where "valid" was defined in Lab_8.

Field 2 – contains a dollar amount corresponding to the total amount paid for the bill, including the tip and sales tax. The format of this field will be $X.DD where X may be any number and D is a digit. For example $19.95.

Field 3 – contains the dollar amount of the tip that was paid as part of the amount in Field 2. Note that the amount of the tip is not taxable, whereas the remainder of the bill is taxable. The format of this field will be $X.DD, for example $12.50.

Field 4 – the number of people in the party, for example, 2.

Field 5 – a string containing comma separated information about the meal purchased; the only important point is that the string may or may not contain the word "wine". An example is: "lunch, tea, water, soda", another example is "dinner, wine, water".

If the credit card number in Field 1 is not a valid number, your program must write the entire line from the *receipts* file into a second text file that has a name of the form: badnumbers-mm-dd-yy.txt where mm-dd-yy corresponds to the same date as the date in the file name from which the line was read. The rest of the line will be processed like any other line.

Here is an example of the desired output for 10-28-13. In addition to what you see here, five lines were written to badnumbers-10-28-13.txt:

```
What receipt file would you like to process? receipts-10-28-13.txt

Deacon Dew Drop Bistro - closing stats for 10-28-13

Total take: $ 6533.47
 Cash take: $ 2040.47
 Card take: $ 4493.00

Tips to disburse: $ 830.70

Tax owed to NC: $ 373.08

There were 549 patrons who spent an average of $11.90 each.

Bills with wine averaged a 17.4% tip.
Bills without wine averaged a 13.9% tip.

There were 5 bad credit card numbers.
```

Details:

- The program should ask the user for the name of a *receipts* file which will include a date in positions 9 – 16.
- The first line of output is a label identifying the restaurant and the date.
- The "Total take" is the sum of all the amounts paid, including the amount paid in cash and the amount charged on a card (even if the card number isn't valid).
- "Tips" is the sum of all the tips.
- The "Tax owed" is the sum of the tax added to each bill (but not the tip) as a 7% sales tax. Your program will need to figure out what tax was charged. This is simple arithmetic. Do the arithmetic in a function with the following heading: **`def computeTax( rate, total ).`** The function should return the amount of tax that was charged, given a total bill and a tax rate. For example computeTax(0.05, 100) should return 4.762.
- You'll need to know the total number of patrons served to compute the average charge per patron.
- The manager believes that diners who buy wine are better tippers. Gather some data to help her decide if that's true or not.
- ++ + Extra credit: The manager also believes that large parties (more than 4 patrons) tend to be better tippers. Gather some data to help her decide if that's true or not.

Deliverable:

- When your program works correctly upload it to the Assignment section of Sakai. Have fun!

SCORING RUBRIC:

30 pts – Program compiles and executes
23 pts - Program computes each of the required values correctly
5 pts – Progam correctly uses a function to compute the amount of tax owed on the bill
7 pts – Program correctly identifies invalid credit card numbers and writes the associated lines to a file with the correct name.
15 pts – Program does not create run time errors
20 pts – Program uses meaningful variable names, is well documented, well formatted, and readable