**ENGR 102**

**Lab # 10Bonus (Individual Activity) [100 Points]**

You are to write the following programs, each of which should be done individually. However, you may talk with others in lab about how to go about doing each of them.

**Activity #1: [40 points]**

One of the most common ways that data can be stored to be loaded in a spreadsheet or other similar table is with a CSV (Comma Separated Value) file. These often are given a .csv extension. A csv file is a way of representing a table in a file. Each line represents a row of the table, and the cells in each column are separated by commas. CSV files can usually be read into spreadsheet programs (such as Excel), and most spreadsheets can output their data in a CSV format (sometimes called “comma delimited” format). You are going to practice writing and reading CSV data directly.

Write a program that will save, to a file, a list of amortized values for a loan. Specifically:

* 1. Ask the user for a file name for saving the data.
  2. Ask the user for the amount of the loan (), the number of months () over which the loan will be repaid, and the annual interest rate (). (Note that should be a decimal number, not a percentage: *0.025* not *2.5%*)
  3. Calculate the monthly payment () as shown below. Note:
  4. Check your calculations: For the monthly payment should be
  5. For each month, calculate the accrued interest by multiplying the beginning balance by . Then calculate the ending balance by subtracting the payment from the beginning balance and adding the accrued interest to that figure.
  6. For each month, write to the output file the month number, the total amount of interest accrued so far, and the amount remaining on the loan, separated by commas.
     1. Start with month 0, when there is no payment and no interest, with month 1 being the first payment and first interest accumulation
     2. The loan will eventually be paid off, so write out values until the balance is $0.00.
  7. Be sure that you write out column headers for the table. It is permissible to include the header strings as comma-separate data on the first line of the file.

Note: if you write your .csv file correctly, you should be able to open it in a spreadsheet program that can read .csv files. You can also check the values themselves using the PMT() function in Excel.

**Rubric: [40 points]**

[5points] – Loan information is prompted for and read in

[5 points] – File name is read in, .csv is added, and this is used for saving information

[10 points] – All data is saved to a file, in appropriate CSV format.

[3 points] – For one year, change in loan is determined correctly.

[5 points] – Change in loan is calculated correctly over all years

[4 points] – If loan is decreasing, all payments are shown

[4 points] – If loan is not decreasing, 30 payments are shown.

[4 points] – Code is commented and reasonably organized

**Activity # 2: [60 points]**

On the class website is a CSV file containing weather data from Coulter Field (in Bryan) for 3 years (1 day is missing for some reason!); the data was taken from Weather Underground (wunderground.com). There are different versions of the file for Windows and Mac; the only difference is whether the end of a line contains just a new-line character, or both a new-line and a carriage return (you don’t need to worry about the difference). Open the file in any text browser and you should see what it is. Note that the first line of the file contains the column headers explaining what each column is.

Download the file to your system, and write a program that will read in the file and do the following:

1. Output the maximum and minimum temperature seen over the 3 year period
2. Output the average daily precipitation
3. Pick any 3 other “interesting” data analysis questions (of your choice) and output the answer to that question. For at least one, make use of the date information. Here are some ideas, but you can pick whatever you want:
   1. For some particular day, such as December 25, find the maximum and minimum temperatures reached among the 3 years of data.
   2. For some particular month, such as July 2015, calculate the average high temperature.
   3. Calculate how frequently the pressure increases from one day to another vs. how frequently it decreases.
   4. Calculate the percentage of days when the humidity was above some value, like 90%.
   5. Calculate the mean and standard deviation of precipitation levels.

Be sure to include a descriptive sentence for what you are printing out in each case.

Note that the “interesting” analysis questions you choose should be different analyses from each other; for example, you should not find the min/max temperature for just 3 different dates, or find the min/max pressure for December 25, since those are essentially the identical computation.

**Rubric: [60 points]**

[15 points] – File is opened, and all data is read in to program.

[5 points] – Min and max temperatures over all dates is determined

[5 points] – Average daily precipitation is determined

[10 points each, **30 points total**] – Three additional “interesting” analyses are performed.

* Subtract 5-7 per analysis if the analysis is just a tiny variation on a previous one
* Subtract 3-5 per item if the analysis is uninteresting/too trivial
* Subtract 10 total if no “date” is used in any part of the computation

[3 points] – All information is output with descriptive statement about what is computed.

[2 points] – Code is commented and reasonably organized