**PYTHON PROGRAMMING**

Project

### **Final Project Steps**

1. Write a program to draw a quiz score histogram.
   1. Your program should read data from a file, where each line of the file contains one number in the range 0 to 10.
   2. Your program must count the number of occurrences of each score and then draw a vertical bar chart for each possible score (0 to 10) with a height corresponding to the count of the number of lines with that score.
      1. For example, if 15 lines contained an 8, then the height of the bar for 8 should be 15.
         1. Hint: use a list that stores the count for each possible score.

1. Write a program that includes the following three functions. (Refer to exercises 11, 12, 13, and 14 of Chapter 6.)
   1. Functions
      1. squareEach(nums)
         1. In this function, nums is a list of numbers. It modifies the list by squaring each entry.
      2. sumList(nums)
         1. In this function, nums is a list of numbers. It returns the sum of numbers in the list.
      3. toNumbers(strList)
         1. In this function, strList is a list of strings, each of which represents a number. It modifies each entry in the list by converting it to a number.
   2. Use the functions above to implement a program that computes the sum of the squares of numbers read from the file.
      1. You should create and save a file with one line and a list of numbers, each separated by a space.
         1. An example would be a file with the following contents: 13 34 14 53 56 76.
      2. Your program should prompt for a “file name” (the name you used when you saved the file) and print or display the sum of the squares of the values in the file.
         1. Hint: use readlines().

1. Write a program that computes the sequential day of the year (365 or 366). (Refer to exercises 11, 12, and 13 of Chapter 7.)
   1. The program should accept a date as month/day/year (mm/dd/yyyy).
   2. Verify that what is entered is a valid date.
      1. If what is entered is not a valid date, display the message “Not a valid date.”
   3. Compute the day of the year using the steps below, using int arithmetic.
      1. dayNum = 31(*month* – 1) + day
      2. If the month is after February subtract (4*month* + 23)/10.
      3. If it’s a leap year (as explained in the referenced exercises) and after February 29, add 1.
   4. Print or display the computed day of the year.

1. Write a program that computes the fuel efficiency of a multi-leg journey. (Refer to exercise 9 of Chapter 8.)
   1. Prompt for the starting odometer reading.
   2. Get information about a series of legs.
      1. For each leg, the user enters the current odometer reading and the amount of gas used (separate by a space).
      2. The user signals the end of the trip with a blank line.
   3. The program should print or display the miles per gallon (MPG) achieved on each leg and the total MPG for the trip.