

## CSC 4800 Python Applications Programming

### Lab #1 – Quiz Data Bar Chart Analysis

Due: Wednesday, January 11, 2017

The purpose of this assignment is a warm-up exercise in Python programming.

(similar to Lab 1 in CSC 2430)

Be sure to implement a clean, well-designed, and commented Python program solution.

Every module/program that you write must have a docstring `""" ... """` heading at the beginning that identifies you and gives a brief synopsis of the purpose of the program. Each function must have a docstring comment on the line following the `"def"` states the purpose of the function, a brief description of the input parameters required by the function, and a clear description of what values the function returns, if any.

---

Design, implement and test a Python 3 program that inputs and stores a collection of data values representing Quiz scores and the number of students that received each score. The input data consists of a collection of pairs of int data values:

- the first integer value in each pair is a quiz score value  
(values between 0 – 50, e.g., 51 possible quiz score values)
- the second integer value in each pair is a frequency count of the number of students that received that quiz score value (e.g., frequency count values are  $\geq 0$ )

(that is, `ScoreFrequency[ k ]` is the number of students that received quiz score `k`).

The input data set is a variable-length sequence of these pairs of int data values, one pair per line. For example, a very small input data set might look like:

```
42 1
35 5
50 2
41 6
42 3
48 4
```

which would result in the `ScoreFrequency` collection of up to 51 values being all zeros except for the indicated score values 35, 41, 42, 48, and 50, which have the associated frequency values indicated. If the data set contains more than one pair of (score, frequency) values for the same quiz score, then add the additional frequency value to the appropriate array entry (for example, if this data set also contained another pair of values `" 42 3 "`, then the `ScoreFrequency[42]` entry would be 4, →the initial 1, plus the additional 3 from the second pair of values for quiz score 42).

The program must prompt the user to enter a filename, and read the input data set from this file. Once the data is input and stored in a dataset collection, analyze the dataset and determine the lowest quiz score that any student received, the highest quiz score that any student received, and the largest frequency count for any of the quiz scores:

(for the small example, `lowestscore = 35`, `largestscore = 50`, and `largestfrequency = 6` ).

Then produce a console output report with three sections:

1. First, echo the list of lines in the input data, followed by the 3 values for the smallest score, largest score, and largest frequency count.

- Second, print out the sequence of values in a "bar chart" table and graph format that identifies the quiz scores and associated frequency counts, one score per line. Each line of output must also contain a horizontal bar string of asterisk characters (\*) representing the frequency count. Only output the sequence of score entries from the lowestscore up to the largestscore range, including all possible scores in this range (even scores with 0 frequency count).
- Third, output another readable "bar chart" graph in a sideways arrangement where the quiz score values (from lowestscore to largestscore) appear across the page and the asterisk-bars are displayed vertically down the page on consecutive lines.

Example output (one long report, split up for printing in this lab assignment handout):

```
Welcome to the Quiz Score Frequency Analyzer, written by Mike Tindall
Enter input data filename: quizdataset1.txt
Reading file 'quizdataset1.txt' input:
42 1
35 5
50 2
41 6
42 3
48 4

The smallest score value is 35
The largest score value is 50
The largest frequency count is 6

---Input Data---
Score: Frequency Bar Chart

35: 5 *****
36: 0
37: 0
38: 0
39: 0
40: 0
41: 6 *
42: 4 *
43: 0
44: 0
45: 0
46: 0
47: 0
48: 4 *
49: 0
50: 2 **

Frequency: Score Bar Chart

^ 6: *
^ 5: *
^ 4: * *
^ 3: * *
^ 2: * *
^ 1: * *
-----:-----^-----^-----^-----^-----^-----^-----^-----^-----^-----^
Score: 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

Dataset 1	
42	1
35	5
50	2
41	6
42	3
48	4

Dataset 2	
35	5
50	2
41	6
42	3
48	4
23	1
42	2
36	1
37	3
39	1
43	8
44	12
45	4
46	5
47	9
49	6

Turn in printed assignment in class.

→ Be sure to use "Courier New" fixed-pitch font for source and output listings.

- Print out of source code.
- Two printouts of sample execution runs:
  - the first run using the short example 6-line test Dataset 1
  - the second run using the longer Dataset 2