

Course: CISC 106: General Computer Science, Honors section

Semester: Fall 2008

Professor: Chandra Kambhamettu

Due: Parts a,b,c due on Dec. 1 by 10pm, Complete program due on Dec. 8, by 10.00PM.

Programming Assignment 3 - Student Score Statistics

1 Academic Dishonesty

Please be sure to re-read the syllabus on Academic Dishonesty. For all programming assignments, students must work INDEPENDENTLY.

2 Grading

80% of the grade depends on your program's correctness and 20% of the grade depends on your program's readability. Your program style should conform with the programs shown in the textbook. Use meaningful variable names and provide indentation to improve readability of your program.

3 Objectives

- Write a pseudo-code showing the list of steps to take for solving the given problem. In this process, you should identify the functions that are to be used in the program.
- Develop a C program based on the designed algorithm.
- Correct the program for any syntax errors.
- Test the program for any logic errors.

4 Assignment

4.1 Overview

You will be writing a program that reads a table of data from a file that user is prompted for, where each row corresponds to a student's score in a given class. Similar to the present class, the student has scores for i) 1 HW (averaged out already), ii) 12 Labs, iii) 3 Programs, iv) 2 Mid-terms, and v) 1 Final. The input will first have the student-lastname, the student-status, HW (1), labs (12), programs (3), mid-term scores, and the final score. The student-status shows year of the student: 1 = freshman, 2 = Sophomore, 3 = Junior, 4 = Senior. There are total 80 students in the class, so there are 80 rows and 20 columns in the table.

You need to compute various student statistics, including a) student average of labs, b) student average of programs, c) weighted average of student's total score (out of 100), d) student's grade in the class, and e) sorted student data. Menu-interface should be provided so that one can select a particular student-lastname, and then ask for any of the above four choices. Student's grade should be calculated using the table given.

You also need, `statistics` which would let you calculate additional statistics. This additional statistics will produce the following information for students belonging to all years, freshman year, sophomore year, junior year and senior year (categorized individually): i) high, low and mean of the weighted-total-score, ii) high, low and mean of lab-average, and iii) high, low and mean of program-average, iv) histogram of labs, programs, exams. Histogram is nothing but the frequency (number of) students obtaining a given score. You can use the C-language `hist.c` presented in the class, which displays '@' character (you may change to any other character).

4.2 Program Design

The program design should make extensive use of functions and arrays. Here is an outline of what the program should do.

1. Invoke a function `show_instructions` which displays a welcome message and general instructions to the user.
2. Prompt the user for the input: a, b, c, d, e, s, q (a: student lab average, b: student program average, c: weighted total score, d: student grade, e: sorted data, s: statistics, q: quit). Invoke the functions appropriately.
3. Use arrays in order to store the entire data.
4. The input will be in a file, containing the table of scores of 80 students; sample file will be provided.
5. Option e would prompt for a filename to write.

4.3 Sample Run

```
-----  
Welcome to the Student Score Statistics!! This program allows you to find  
selected statistics of a student's score.  
-----
```

Type any of the following alphabet now.

```
a for student-lab-average  
b for student-prog-average  
c for weighted-total-score  
d for overall-student-grade  
e for sorted data  
s for statistics  
q for quit
```

Type-in now: a

Input Student-name: XYZ

The student's (name : XYZ) lab average is, 86.3.

Sending back to the menu.

a for student-lab-average
b for student-prog-average
c for weighted-total-score
d for overall-student-grade
e for sorted data
s for statistics
q for quit

Type-in now: d

Input Student-name: XYZ

The student's (name : XYZ) overall grade is, B+.

Sending back to the menu.

a for student-lab-average
b for student-prog-average
c for weighted-total-score
d for overall-student-grade
e for sorted data
s for statistics
q for quit

Type-in now: e

This option is for sorting students and printing in a file, upon:
(i) lab-average, (ii) program-average, (iii) mid-term average,
(iv) final and (v) weighted-total-score

Type-in one of the options (i)-(v): iii

You have selected sorting of Student Data upon mid-term average.

Type-in the file to write: student_mid_avg

Print Completed.

Sending back to the menu.

a for student-lab-average
b for student-prog-average
c for weighted-total-score
d for overall-student-grade
e for sorted data
s for statistics
q for quit

Type-in now: q

Thank you for using SSS (Student Score Statistics).

5 What to Hand In

Run your program for each menu operation and make a diary file for grading. The data can be obtained from `student.data` file, provided to you. For those doing extra credit, run the program on each extra option created for the graphics.

6 Extra Credit (10 points)

Give options for plotting curves for each of the option accordingly. For example, option (c) would have x-axis as each student (1 to 80) and y-axis as the weighted total score.

7 Weights and grades To use

15% Mid-term Exams
30% Final Exam
5% Homeworks
20% Labs
30% Programming Assignments

100%

A	93	\leq	g	
A -	90	\leq	g	< 93
B +	87	\leq	g	< 90
B	83	\leq	g	< 87
B -	80	\leq	g	< 83
C +	77	\leq	g	< 80
C	73	\leq	g	< 77
C -	70	\leq	g	< 73
D +	67	\leq	g	< 70
D	63	\leq	g	< 67
D -	60	\leq	g	< 63
F			g	< 60