BIRZEIT UNIVERSITY

Engineering and Information Technology Faculty
Computer Science Department
Comp231-Advanced Programming

Assignment 2:

Deadline:Thursday 11/11/2021 before midnight.

Objectives:

1. String , classes and objects (Chapter 1- Chapter 10)

Note: YOU CAN'T USE ANY OTHER CONCEPT OUTSIDE THIS CHAPTERS (WILL NEVER BE GRADED).

Problem: Graduation Seminar Presentation

Your course instructor tired to sort the students in the class in regular alphabetical order, since this means that certain **students (*first name, last name, id, date of birth, grade,email*)** always go first for presentations. He/She has an idea to mix up the order of the students by sorting them in a different manner. In particular, he/she *doesn't care about* the order of the letters in the last name of a student. Instead, he/she will put the student who has the *highest number of A's in his or her last name first*. If there is a tie between two students based on this value, she'll compare the number of B's in their last names. If this is tied also, she will go on to C's, then D's, etc. The second student will be found using the same method, and so on until all of the students have presented. Your job is to sort a list of student names based on this criterion.

It is guaranteed that no class has two students who have names that are anagrams of one another; thus, there will not be any ties between two students (an anagram is a different arrangement of the same exact letters).For example, "abcd" and "dabc" are an anagram of each other.

Write **a method** to generate an email automatically for student by take the first letter for his/her name followed by last name followed by @ special character and then ritaj server name with extension "birzeit.edu", i.e *nkhader@ritaj.birzeit.edu*

The first line of the input consists of a single positive integer, *n*, representing the number of classes . Each class contains a single positive integer, *m* $(1 < m < 100)$, which represents the number of students in that class. Each of the following *m* lines will contain (first name, last name ,id and date of birth) . **(you have to take care about cases of letters**)Then, for ordered list name according *to this scheme, instructor can record the grade of presentation for each student.*

- Use proper constructor (s), setter(s) and getter(s) for defined student class.
- Write a test driver that contains:

1. a method to calculate the average for each class.
2. a print method to will display the list of student and average at screen , i.e (student name,email, grade, and AVG of class)

- Be aware of all concepts of object oriented you have been learned in these chapters.

Sample Input

Enter Number of classes: 2

Enter Number of students in class# 1 : 2

AHMAD BIDAS
SAMI JAWAD

Enter Number of students in class# 2 : 3

SAMER WALAWI
LEEN MAHAREEQ
HANEEN DARAHMAD
……
….
..

Sample Output

1. Print list of student sorted for class #1 ,class #, etc
2. **The Instructor can enter *the grades*** for those classes.
3. After grades were entered, Print the following results at screen:

Class #1 ordering

SAMI JAWAD , *email:sjawad@ritaj.birzeit.edu,78*
AHMAD BIDAS, *email:abidas@ritaj.birzeit.edu ,90*
……
….
..

AVG of class #1 :82.65%

Class #2 ordering

HANEEN DARAHMAD, *email:hdarahmad@ritaj.birzeit.edu,85*
LEEN MAHAREEQ, *email:lmahareeq@ritaj.birzeit.edu,76*
SAMER WALAWI, *email:swalawi@ritaj.birzeit.edu,82*

AVG of class #2 :78.75%
……..
…..
…

Specification Submission:

1. *Online through ITC.*
2. *What to submit: Your own well-structured and **well-commented** JAVA files (.java) .**ONLY class and driver as .java format)***

Grading policy and general notes on the Assignment:

1. Your application should have all functionalities working properly. Twenty percent of marks will be graded for the functionality of the assignment.
2. The following notes will make up the remaining 10 marks of the grade:
   - **a.** There has to be adequate documentation and comments in the code(i.e., functions, loops, etc.);
   - b. Your code should follow the code convention (i.e., spaces, indentations,etc.); and
3. Any plagiarized code will not be marked.
4. ANY LATE Assignment will never be accepted for any excuse.

Types of cheating:

1. **outsource, like books, internet.**
2. **classmate.**
3. **Facebook groups or from any social media.**
4. **Chegg website or other similar .**

Good Luck!!