Lab 04 Specification – Explore Data Processing using Two Dimensional Arrays and Array of Objects
Due (via your git repo) no later than 8 a.m., Wednesday, 13th Mar 2019.
50 points

Lab Goals

- Practice working with two-dimensional arrays and array of objects.
- Implement a multiple choice test Grader tool
- Answer a few questions to test your knowledge about the class content to-date

Suggestions for Success

- Take a look at the suggestions for successfully completing the lab assignment, which is available at: `https://www.cs.allegheny.edu/sites/amohan/resources/suggestions.pdf`

Learning Assignment

If you have not done yet, you should read the following to do well on this assignment. It is also highly important to review the code from class discussions, lecture slides, and class notes to do well in this lab.

- GT chapter 03 - Section 3.1

Assignment Details

In this lab, we will focus on two-dimensional arrays and array of objects, which are the two current talking points from our in-class discussions. Although we started discussing Recursion during our most recent class, we need to spend some more time in the topic before including Recursion in a lab. Thereby, this lab does not cover Recursion. This lab has a to-do and to-think component in it. The to-think part of the lab is an opportunity provided for you to think about the different type of data structure design that is required to solve data driven problems efficiently. The lab is also designed with an intention to relax the workload provided for the week, as there is an EXTRA credit challenge due by next Tuesday. So, I encourage you to spend your extra time wisely and provide the best solution for the Design #3 challenge presented to you in class and make use of the EXTRA credit.

A two-dimensional array share similar properties as a one-dimensional array in the context of homogeneous nature (all cells in the array need to be of same data type!). However, an interesting addition to a two dimensional array is that the array structure can contain more than one row and more than one column. This interesting addition allows a programmer to represent multi dimensional data inside the structure.

The major part of this lab is to practice working with a two-dimensional array and also combine your understanding of one-dimensional array from the previous lab to implement a multiple choice test grading tool. A thinking exercise along with the core part of the lab is to think out of the box and design an efficient data structure that address the same requirement of the lab (maybe using Array of Objects). Additionally, you will also answer a few short questions at the end to test your knowledge about the class content to-date.

So how to compile the program

At this point we should all be familiar with the compilation of Java programs using **javac** command. Since, there is a particular directory structure associated with the lab and to make the compilation process simple, there is a compile bash script shipped with the lab repository. In order to compile your programs, make sure to place all your Java files under the src directory, Then, open the terminal and navigate to the lab repository and type in:
"./compile.sh"
This should compile all your files and automatically place their class files in the classes sub directory located in the lab repository.

So how to execute the program

At this point we should all be familiar with executing Java programs using **java** command. Since, there is a particular directory structure associated with the lab and to make the execution process simple, there is a run bash script shipped with the lab repository. In order to execute your programs, make sure to open the terminal and navigate to the lab repository and type in:
"./run.sh Grader"
Here Grader is the name of the class file that is to be executed. If you have a different class file that needs to be executed, then specify that class name while executing the run.sh bash file.

To maintain a uniform standard for all lab submissions, it is recommended to use a Linux environment in our CS-101 class and lab sessions. As you may note, our CS department had also installed Linux based OS in all our lab machines to maintain the uniform standard across the board in our learning environment. In order to align with our learning environment standards, the compilation and execution utility [compile.sh and run.sh] provided with the lab are based on strictly Linux environment. This does not mean a student is not allowed to use other operations systems such as Windows, Mac, and so on. The student has the complete freedom to chose any Operating System of their choice to implement the lab, but it is recommended to use Linux based OS similar to our lab machines. If a student uses a different operating system other than Linux, then the student is required to include all the necessary details of the compilation and execution process clearly in the READ ME file. Additionally, it is the student's responsibility to make sure and test that the code produced is portable and executable in a Linux environment before submission. In order to maintain the uniform standard, the grading process will be done by the Professor using a Linux based machine provided by the college.

Simple Grading Tool (40 points)

In a recent class, we had discussed setting up a two dimensional array, and seen how an iterative block of code can be set up using a nested for loop in order to do the data processing. The grand idea behind the data processing is to set up the outer for loop to iterate through each rows and the inner for loop to iterate through each columns in the array structure.

The major highlight of implementing this Simple Grading Tool is to program the requirements outlined below to grade multiple choice tests. Let us suppose that there are:

1. eight students
2. ten questions
3. the answers are stored in a two-dimensional array
4. each row in the two-dimensional array records a student's answers to the questions
5. The key [correct answers] is stored in a one-dimensional array.

For example, the array shown below stores the test:

Students' Answers to the Questions:

	0	1	2	3	4	5	6	7	8	9
Student 0	A	B	A	C	C	D	E	E	A	D
Student 1	D	B	A	B	C	A	E	E	A	D
Student 2	E	D	D	A	C	B	E	E	A	D
Student 3	C	B	A	E	D	C	E	E	A	D
Student 4	A	B	D	C	C	D	E	E	A	D
Student 5	B	B	E	C	C	D	E	E	A	D
Student 6	B	B	A	C	C	D	E	E	A	D
Student 7	E	B	E	C	C	D	E	E	A	D

The one-dimensional array shown below represent the key that includes the correct answers to the questions outlined above:

Key to the Questions:

	0	1	2	3	4	5	6	7	8	9
Key	D	B	D	C	C	D	A	E	A	D

Data Processing Outline: A critical component of implementing this part is to process both the 2D [test] and 1D [key] arrays and perform a series of comparisons to evaluate the number of correct answers, incorrect answers, computing the final score, and finally computing the class average. In your implementation, you may assume that student id is the same as the **index** value of the row representation in the test array.

In order to process the data as required, the outline of the core logic is stated below:

1. Set up the input arrays for both the 2D and 1D arrays by using an appropriate data type.
2. Iterate through each student's answer and compare the student answer with that of the correct answer provided in the key array.
3. If the answer matches, then increment the correct count
4. Else increment the incorrect count
5. Compute the class average by summing up the total score of all the student's and by finding the average.
6. Finally output the total number of correct answers, incorrect answers, the total score of the student, and the class average based on user provided student id.

An expected sample output from your program execution should look similar to the output shown below:

”Hello student, what is your student id? 4”
”Congrats on taking the multiple choice test. You had received 8/10 with 8 correct, 2 incorrect answers and the class average is 6.375!”

Input requirements:

1. It is acceptable to make the assumption that both the 2D and 1D input arrays strictly contains char values.
2. It is acceptable to directly hard code the content in your implementation from the arrays shown in the figure above. I don’t expect you to perform any additional data validations in your code, as the input is hard coded in your program.

To Think: There is no deliverable expected for this part of the lab. This is an opportunity for you to think through and come up with a design solution that involves identifying a suitable data structure using the **”Array of Objects”**. In the actual lab to-do part, you had used a two-dimensional array for the tests and a one-dimensional array for the keys. Additionally, I expect you to use one or more arrays internally in your code, to set up the requirements and the logic part of the lab. So now, maintaining a number of distinct arrays like that is not an efficient way of doing coding. Instead, the efficient way is to use one solid data structure that internally holds different members including arrays and nicely encapsulate all the details from an external world. Think of a data structure (along with the lines of Student and Student1 data structure we had developed in class) to meet all the requirements of the lab in a more elegant and efficient way.

Additional Questions (10 points)

Please answer the following questions thoroughly:

1. Let us suppose that you are creating an Array of objects? The object values can be assigned using a constructor approach or by using the setter approach. Explain the key difference between the constructor approach and the setter approach?
2. We discussed in class that Arrays.toString() is a utility function that lets you print the Array contents in the console. How is this approach different from setting up an iterative code to print the array contents by yourself? What are some limitations and advantages of using the utility function such as Arrays.toString()?
3. Although we had not fully discussed Recursion, I think it is important for you to reflect your understanding about Recursion from our discussion so far. How is recursion different from an iterative data processing technique? What are some pros and cons on both sides? Note: We will discuss this in detail in our next class session. But, a proper reflection from our previous class should help you understand the comparison in our next class better.

Submission Details

For this assignment, please submit the following to your GiHub repository by using the link shared to you by the course instructor:

1. Your source code for `Grader.java`
2. Sample output runs for `Grader.java` in PDF format. A screenshot of the output from your terminal can be included.

3. The answers to the questions from the “Additional Questions” section in a PDF format.
4. A readme file explaining how to compile and run your program. In addition, include a detailed self reflection of the lab exercise in the readme file. The self reflection may include answers to questions such as:
   - ”What challenges did you face in this lab?”
   - ”What did you like in this lab?”
   - ”What did you not like in this lab?”
   - ”How long did you spend on this lab assignment?”
5. It is highly important, for you to meet the honor code standards provided by the college. The honor code policy can be accessed through the course syllbus. Make sure to add the statement ”This work is mine, unless otherwise cited.” in all your deliverables such as source code and PDF files.

Grading Rubric

1. There will be full points awarded for the lab, if all the requirements in the lab specification are correctly implemented. Partial credits will be awarded if deemed appropriate.
2. Failure to upload the lab assignment code to your git repo, will lead you to receive ”0” points given for the lab submission. In this case, there is no solid base to grade the work.
3. There will be no partial credit awarded if your code doesn’t compile correctly. It is highly recommended to validate if the correct version of the code is being submitted before due date and make sure to follow the honor code policy described in the syllabus. If it is a late submission, then it is the student’s responsibility to let the professor know about it after the final submission in github. In this way, an updated version of student’s submission will be used for grading. If the student did not communicate about the late submission, then automatically, the most updated version before the submission deadline will be used for grading purposes. If the student had not submitted any code, then in this case, there is 0 points to the student automatically for the lab work.
4. If you needed any clarification on your lab grade, talk to the instructor. The lab grade may be changed if deemed appropriate.