New York University
Computer Science Department
Introduction to Software Development
Dr. Anasse Bari
Homework#6: *Learning Multi-Dimensional array, Stack Data structure and (Part. I) Intro to Object Oriented Paradigm*
Deadline: *See NYUclasses for the deadline, 15% off per day after the deadline (3 days maximum).*

Learning Objective:

- Learning and practicing multi-dimensional arrays
- Learning how to represent a stack data structure as an array
- Getting Started on Object Oriented Design

Read the guidelines bellow carefully to avoid receiving a zero grade on the HW:

- Please use the following naming convention "ExerciseX.java", where X is the number of the exercise. Please apply this convention to all coming homework assignments during the semester. Also please attach the answers and include them into HW's zip file.
- Compile and run the program for each exercises.
- Comment Your Code (at least 5 comments per exercise)
- Make an archive (zip file or compressed file) with all the **java files (the .java files NOT the .class files)** and post it on NYU Classes (StudentName_Homework_X.zip)
- It is your responsibility to make sure that the Zip files has your actual files. You may send the file to yourself by email to double check that is the actual file before you upload on NYU classes.
- **If the graders cannot open the file, you will receive a grade of zero.**
- **If you send the .class files instead of the .java files (source files) you will receive a zero.**
- Please **do not send the entire project**, the Java file for each exercise is all that we need.
- An act of cheating will be severely addressed with an immediate zero on the homework and a report to the academic advisor and the administration.
- You will automatically lose 50%of the points for an exercise if the program does not compile and run correctly.
- The number of points for each questions is being specified.
- Plagiarized assignment will get a ZERO grade. You cannot change the variable names of other student's solution and submit it as yours. The program structure of other students must not match yours. Every student has to come up with his/her own solution. Any cheating (e.g. copying from internet without citing sources) is a serious violation of the University student code.

- If you have questions, please ask them on PIAZZA.
- Make sure you visit the tutors or the instructor if you have any questions.

Guidelines:

- Compile and run the program for the input parameters specified for each exercises.
- Please use the following naming convention "*ExerciseX.java*", where X is the number of the exercise. Please apply this convention to all coming homework assignments during the semester.
- Make an archive (zip folder – Compressed Folder) with all the **java files** and post it on NYU Classes *(Homework_X.zip)*
- **Please comment you code. Explain major lines in your program.** Comments are should be brief and descriptive.
- Please **do not send the entire project**, the Java file for each exercise is all that we need.
- If you do not follow these guidelines, we will take points off.
- If you send .class files, you will get a zero.
- Late submissions: -5% on each day.

Exercises

Pre-requisites. Read chapters 8, 9 and 10 and make sure you attend ALL lectures.

Exercise.1 (20pts) (Sum elements column by column) Write a method that returns the sum of all the elements in a specified column in a matrix using the following signature:

public static double sumColumn(double[][] m, int columnIndex)

Write a test program that reads a 3 by 4 matrix and displays the sum of each column. Here is a sample run: (the matrix has to be entered by the user as shows bellow (row by row))

Enter a 4 by 4 matrix row by row:

1.5 2 3 4
5.5 6 7 8
9.5 1 3 1

Sum of the elements at Column 0 is 16.5
Sum of the elements at Column 1 is 9.0
Sum of the elements at Column 2 is 13.0
Sum of the elements at Column 3 is 13.0

Exercise.2 (20pts) (Sum of the major diagonal in a matrix) Write a method that sums all the numbers in the major diagonal in an n x n matrix of double values using the following header:

public static double sumMajorDiagonal(double [][] m)

Write a text program that reads a 4 x 4 matrix and display the sum of all its elements on the major diagonal. Here is a sample run: (the matrix has to be entered by the user as shows bellow (row by row))

Enter a 4 by 4 matrix row by row:

1 2 3 4.0
5 6.5 7 8
9 10 11 12
13 14 15 16

Sum of the elements in the major diagonal is 34.5

Exercise.3 (30pts) Guess the Capitals. Write a program that repeatedly prompts the user to enter a capital for state. Up receiving the user input, the program reports whether the answer is correct. Assume 50 states and their capital are stored in two-dimensional array. The program prompts the user to answer all states' capital and displays the total correct count. The user answer is NOT case-sensitive.

Your two-dimensional array should be stored like:

Alabama Montgomery
Alaska Juneau
Arizona Phonix
… ….

You can find the full list here: http://state.1keydata.com/state-capitals.php

Here is a simple run:

What is the capital of Alabama?
Montecarlo

The correct answer should be Montgomery

What is the capital of Alaska?
Juneau
You answer is correct.

What is the capital of Arizona?
…

The correct count is 35

Exercise 4 (10pts) Stack Data Structure implemented as an Array

A stack is a data structure that consist of a collection of elements structured in a last-in-first-out structure (LIFO). In this exercise, you will need to implement the concept of a stack (LIFO) using an array. You will need to create a method called push that add elements to the array in LIFO manner, and a method called pop that return the last element that was pushed (added/entered) to the array.

Here is an example of operations on a stack of characters:

Suppose we have a stack that can hold letters (Characters) , call it stack.

Let us begin with empty stack

```
-----
stack
```

if you perform Push(A), the stack (implemented as an array would look like)

```
-----
| A |  <-- top
-----
stack
```

Again, another push operation, Push(B), the stack would be:

```
-----
| B |  <-- top
-----
| A |
-----
stack
```

Now let's remove an item, letter = Pop( ):

```
-----            -----
| A |  <-- top     | B |
-----            -----
stack             letter
```

Push(C):

```
-----
| C |  <-- top
-----
| A |
-----
```

stack

You can make your own design and assumptions on how to declare the array (global vs local), the signature of the methods, the possible use of a variable that could hold the array actual size, you may want to have a maximum size of the array that the use the wont go beyond… the array should be of type chars and you will need to have at least the method pop and push as explained above.

Create a test program that test the operation of push and pop on your stack. Comment your program.

Exercise. 5 (20pts) Getting started with Object Oriented Design
The Account Class.

Design a class named Account that contains:

1. A private int data field named id for the account (default value 0)
2. A private double data field named balance for the account (default value 0)
3. A private double data field named annualInterstRate that stores the current interest rate (default 0). Assume all account have the same interest rate.
4. A private Date data field named dateCreated that stores the date when the account was created.
5. A default constructor that creates the default account.
6. A constructor that that creates an account with the specified id and initial balance.
7. The accessor and mutator methods for id, balance and annualInterstRate
8. The accessor method for dateCreated
9. A method named getMonthlyInterset() that returns the monthly interest
10. A method named withdraw that withdraws a specified amount from the account.
11. A method named deposit that deposits a specified amount to the account.

Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000 and an annual interest of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest and the date with this account was created.