CSCI 1011 – Lab 11

Learning Outcomes

- Represent collections of data using arrays.
- Solve computational problems using loops and arrays.
- Construct classes and methods that operate on arrays.

Required Reading

Savitch - 7.1-7.2

Instructions

1. Start NetBeans.
2. Create a new project called `Lab11` with a main class called `YournameLab11` with your name.
3. Create a Java class file for a `Course` class.
4. Implement the `Course` class.
    (a) Add two private instance variables of type `String` to store the course code and course title of a particular course.
    (b) Add a private instance variable of type double to store the number of credit hours for the course.
    (c) Add a private instance variable that is an array of `doubles` that represents a collection of student scores.
    (d) Write a four-argument constructor that takes the course code, course title, number of credit hours, and number of students in the course and does the following:
    (e) Use the course code, course title, and number of credit hours to initialize the respective instance variables.
    (f) Use the `new` keyword to initialize the array of scores to a new array whose length corresponds to the number of students in the class.

(g) Write a public void method called `readScores` that prompts the user to enter the score for each student in that course, and stores the score in the corresponding array element. Example input and output might look like this:

```
Enter the course code:
CSCI 4200
Enter the course title:
Principles of Information Security
Enter the credit hour:
3.0
Enter the number of students in the course:
6
Enters scores for 6 students
87
64
96
88
45
73
```

(h) Write a public void method called `displayInfo` that displays the course information in the following format:

```
CSCI 4200: Principles of Information Security (3.0 credit hours)
Class size: 6
Scores: 87 64 96 88 45 73
```

5. Write code in the main class to test the `Course` class.

(a) Write a static method called `createCourse` that does the following:

- Prompt the user for the course code, title, credit hour and class size.
- Create an object of type `Course` using the 4-argument constructor and the values the user provided.
- Return the `Course object`

(b) Add the following code to the `main` method:

- Call the `createCourse` method to create a Course object and store it in a variable.
- Call the `readScores` method to read scores of all students in the course.
- Call the `displayInfo` method to display the results.

(c) Run your program to make sure that it works correctly.

6. Add additional methods to the `Course` class.

   (a) Write a private helper method that calculates and returns the sum of all of the scores in the array.

   (b) Write a public method called `getAverage` that computes the average score using the private helper method to calculate the sum and the `length` instance variable of the array.

   (c) Write a public method called `getMax` that returns the maximum score in the course.

   (d) Write a public method called `getMin` that returns the minimum score in the course.

7. Write code in the `main` method of the main class to test the new methods.

   (a) Display the average score in the course using the `getAverage` method.

   (b) Display the maximum score in the course using the `getMax` method.

   (c) Display the minimum score in the course by calling the `getMin` method.

   (d) Make sure to display exactly 2 digits after the decimal point when displaying each of the computed scores.

   (e) If the input from step 4 is used, the output of the above tests should look like this:

```
The average score is: 75.50
The maximum score is: 96.00
The minimum score is: 45.00
```

8. Hand in the source files, `YournameLab11.java` and `Course.java` to the D2L assignment dropbox called **Lab Assignment 11**.