PROG36859 Object Oriented Programming 2

Sheridan | Get Creative

Section (P02 – 59317)

Fall 2018

Assignment 1

Weight: 10%
Max. Points: 60
Due Date: Sept 28, 2018 11:55 pm

1. (40 marks)
   1.1). Write a program that reads a student's test scores, gets the best score, and then assigns grades based on the following scheme:

   Grade is A if score is : best - 10
   Grade is B if score is : best - 20;
   Grade is C if score is : best - 30;
   Grade is D if score is : best - 40;
   Grade is F otherwise.

   The program prompts the user to enter the total number of students, then prompts the user to enter all of the test scores for all courses taken in the semester for each student , and concludes by displaying the grades. Here is a sample run:

   ```
   Enter the number of students: 4
   Enter 4 scores: 40 55 70 58
   Student 0 score is 40 and grade is C
   Student 1 score is 55 and grade is B
   Student 2 score is 70 and grade is A
   Student 3 score is 58 and grade is B
   ```

   The maximum number of scores can be entered is 5, and the scores must be stored in an Array.

   1.2). computes the standard deviation of the scores in part 1. Here is the mathematical formula for calculating the standard deviation:

$$\text{mean} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

$$\text{deviation} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \text{mean})^2}{n - 1}}$$

The output of your program displays the mean and standard deviation, as shown in the following sample:

```
 scores:  6 3 4 5 2
The mean is 4.0
The standard deviation is 1.58
```

1.3). sort the “scores” array using the Arrays class defined in java.util package;

Programming specifications:

a. You need to implement Five methods. The names of the methods are findBest(..), findMean(…), find standardDeviation(…), outputResult(…), and SortingScores(…). and call these methods in the main method to produce the output.

b. Please add exception handling code to deal with ArrayIndexOutOfBoundsException for all the array manipulations in your program.

2. (20 marks) create a two dimensional array that stores a student’s course records for this semester and output the student final grade for each course as %. Note: Please use a nested for loop to manipulate this two dimensional array.

Example:
Student name: Smith Smith
Grade Info:

Prog1	Midterm mark	Final test mark	Assignment mark	Lab mark
Prog2	Midterm mark	Final test mark	Assignment mark	Lab mark
Info3	Midterm mark	Final test mark	Assignment mark	Lab mark
Syst4	Midterm mark	Final test mark	Assignment mark	Lab mark
Prog5	Midterm mark	Final test mark	Assignment mark	Lab mark

Submission Standards for PROG36859

All submissions for this course must meet the following requirements:

Program Documentation

Programmer Identification

All submissions must include program identification documentation at the top of the source code file according to the following specifications:

- **Main class files** - the file in your project that contains the main() method must include the following documentation in multi-line comment syntax (/* ... */):
- `Name:  [your name here]`
- `Assignment:  [the name/title of the submission]`
- `Program: [the name of the program you're enrolled in i.e. Computer Programmer or Programmer Analyst]`
- `[a blank line]`
  `[brief description of program]`
- **Class files** - classes, abstract classes, or interfaces that model objects or contain libraries of code must include documentation in multi-line comment syntax that identifies the programmer(s), the current source file and all other source files that are included in the project, and which class is the "main()" class, as shown below:
- `Name: [your name here]`
- `      [programmer #2's name, if applicable]`
- `      [programmer #3's name, if applicable]`
- `File:  Employee.java`
- `Other Files in this Project:`
- `  FullTimeEmployee.java`
- `  PartTimeEmployee.java`
- `  CommissionEmployee.java`
- `  EmpGui.java`
  `Main class: Main.java`

Javadoc Comments

All OOP classes submitted must be properly documented using javadoc comments (/** ... */) with proper tags and attributes.

For more information, see Javadoc Comments for more information.

Coding Standards and Conventions

All code must meet the following industry standards.

Electronic Submissions

Electronic submissions are usually uploaded to SLATE.

The policy regarding submitted files is as follows:

- All submissions must be contained within a single ZIP file.
- Your ZIP file must be named **`login_assignmentName.ZIP`** where **"login"** is your actual Sheridan login name and **"assignmentName"** is the actual name of the assignment you are submitting (e.g. Lab1, Assignment2).
- If you are asked to submit **source code only**, then your ZIP file should contain a complete Eclipse project: Find your project's root directory and zip the entire directory into a single zip file. Ask your professor for a demonstration if this is not familiar to you.

Make sure you read your assignment's submission instructions carefully so that you submit the correct files/folders.