University of Windsor SCHOOL OF COMPUTER SCIENCE	COMP-2120, Fall 2024 Object Oriented Programming Using Java LAB-3 TOTAL MARKS: 10

LAND ACKNOWLEDGEMENT
The School of Computer Science at the University of Windsor *sits on the Traditional Territory of the Three Fires Confederacy of First Nations*. We acknowledge that this is the beginning of our journey to understanding the Significance *of the history of the Peoples of the Ojibway, the Odawa, and the Pottawatomie.*

SUBMISSION DEADLINE: OCT-11, 2024

GENERAL INFORMATION

Welcome to the Lab Session of COMP-2120 where you will learn object-oriented programming (OOP) using Java. Please, arrive early to the lab and get settled to start working on the lab. Arriving 20 minutes after the start of the lab will be considered late and can affect your lab participation point. There are eight lab exercises you need to complete throughout the course. Each week on Thursday morning we will publish the lab exercise. You need to upload your code by Thursday, 10 AM in the next week.

LAB GRADING AND ASSESSMENT

You need to explain your code to TAs in the lab to receive the marks. YOU MUST demonstrate your results and explain code to the TAs to receive the marks. Thus, attending the lab sessions is mandatory. Note that your TAs are here in the Lab to help you learn. Feel free to ask questions and talk to them. Remember, the total Labs is worth 16% of your final grade.

Lab-3 Tasks

Part A: Implement in the `Student` Class

1. **Class Definition and Constructor (1 Point)**

   Define a **`Student`** class with the following attributes: **`name`** (String), **`id`** (int), and **`marks`** (float[] for 3 subjects). Then **write a constructor** to initialize these attributes.

2. **Average Marks Calculation (1 Point)**

   Implement a method **`getAverageMarks()`** in the **`Student`** class that calculates and returns the average marks of the student.

3. **Display Student Details (1 Point)**

   Write a method `printStudentDetails()` in the `Student` class to display the student's **name, ID, and average marks.**

4. **Pass/Fail Status (1 Point)**

   Add a method `isPassing()` to the `Student` class that returns `true` if the student has passed all subjects (passing marks >= 50).

5. **Find the Top Scorer (1 Point)**

   Write a static method `findTopScorer(Student[] students)` to find and display the details of the student with the highest average marks.

Part B: Implement in the `Test` Class (another class, where the `main` method is)

6. **Input and Validation (2 Points)**

   Inside the `main` method of the `Test` class, create an array of five (5) `Student` objects.

   For each student, input their `name, id`, and **marks** in three (3) subjects. Ensure that all marks are valid (**between 0 and 100**).

7. **Sorting by Average Marks (2 Points)**

   In the `main` method of the `Test` class, sort the students by their average marks in descending order. Display the sorted list of students using the `printStudentDetails()` method.

8. **Search by ID (1 Point)**

   In the `main` method of the `Test` class, allows the user to search for a student by their ID. If the student exists, display their details using the `printStudentDetails()` method; otherwise, display a message that the student was not found.

WHAT DO YOU NEED TO DO?

1. Complete the program.
2. Attend the lab. Explain the code to TAs.
3. TAs can modify the input to your program.