University of Windsor SCHOOL OF COMPUTER SCIENCE	COMP-2120, Fall 2024 Object Oriented Programming Using Java LAB-8 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: NOV-29, 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 the code to the TAs to receive 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 of eight Labs is worth 16% of your final grade.

Lab-8

In this assignment, you will work with a file containing student data, perform calculations, assign grades, and identify the top scorer:

First, create the input file: The input file, students.txt, contains the following information: (create the file first with the name Student.txt)

ID, Name, Marks1, Marks2, Marks3

101, Alice, 85, 78, 92

102, Bob, 90, 88, 84

103, Carol, 75, 72, 70

104, David, 95, 92, 96

105, Eve, 60, 65, 62

Task 1: Read and Display File Content **(3 points)**

a. Write a method to read the file students.txt.
b. Display the content of the file line by line on the console.

Task 2: Calculate Average Marks **(2 points)**

a. Write a method to calculate the average marks for each student.
b. Write the updated student records, including average marks, to a new file named students_with_avg.txt.

Task 3: Assign Grades **(2 points)**

a. Write a method to assign a grade to each student based on their average marks:
   - A: 90–100
   - B: 80–89
   - C: 70–79
   - D: 60–69
   - F: Below 60

b. Update the students_with_avg.txt file to include the assigned grades.
   Format:
   ID, Name, Marks1, Marks2, Marks3, Average, Grade

Task 4: Find the Top Scorer **(3 points)**

1. Write a method to identify the student with the highest average marks.
2. Append this information to the end of the file students_with_avg.txt.
   Format:
   Top Scorer: [Name], Average Marks: [Average], Grade: [Grade]

Final Output Example (after completing all tasks): The output file **students_with_avg.txt** file should look like this:

ID, Name, Marks1, Marks2, Marks3, Average, Grade

101, Alice, 85, 78, 92, 85.0, B

102, Bob, 90, 88, 84, 87.33, B

103, Carol, 75, 72, 70, 72.33, C

104, David, 95, 92, 96, 94.33, A

105, Eve, 60, 65, 62, 62.33, D

Top Scorer: David, Average Marks: 94.33, Grade: A

OTHER INSTRUCTIONS:

1. **File Handling:**
   - Use BufferedReader and BufferedWriter for file operations.
   - Ensure proper exception handling for file input/output.
2. **Code Modularity:**
   - Divide the code into methods:
     - A method to **read and display the file**.
     - A method to **calculate averages**.
     - A method to **assign grades**.
     - A method to **find the top scorer**.
3. **Error Handling:**
   - Validate data while reading the file (e.g., marks should be between 0 and 100).

WHAT DO YOU NEED TO DO?

1. Complete the program.
2. Attend the lab. Explain the code and the purpose of using different OOP features to a TA.
3. TAs can modify the input to your program.