CITY UNIVERSITY OF HONG KONG

Course code & title	:	EE3206 Java Programming and Applications
Session	:	Semester A, 2021/22
Time allowed	:	2.5 hours

This paper has 4 pages (including this cover page).

---

1. This paper consists of 2 questions.
2. Answer ALL questions in this paper.

This is an open-book examination.

Candidates are allowed to use the following materials/aids:

- *Textbooks*
- *Printed course materials*
- *Approved computer and software*

Materials/aids other than those stated above are not permitted. Candidates will be subject to disciplinary action if any unauthorized materials or aids are found on them.

NOT TO BE TAKEN AWAY

Instructions for Computer Programming Exam

In addition to this exam paper, you are given two program templates and one CSV data file as list below:

- Q1_XXXXXXXX.java
- Q2_XXXXXXXX.java
- athletes.csv

After you download the files from Canvas, you should:

1. Replace the string XXXXXXXX in the class name and file name to your actual student ID e.g. Q1_56781234.
2. Write down your name and student ID on top of each program template.

When you start working, you should:

1. Read the questions on this paper carefully and follow the requirements specified on the program templates. You should NOT assume any data patterns or orders in the CSV data file unless specified explicitly in the question.
2. Write code inside the provided methods ONLY. Do not modify any other parts in the templates, except that you can import the necessary packages.

During the exam, you should NOT:

1. Open or use any unauthorized software such as internet browser.
   (Note that Java API documentation can be accessed within NetBeans IDE via *Window > IDE Tools > Javadoc Documentation*)
2. Open or use any digital reference materials such as lecture notes and program codes.

When you finish the exam, you should:

1. Upload the two completed .java program files (with your answers) to Canvas.
2. Check again in Canvas to ensure you successfully upload the answers.

Question 1 (65 marks)

Since 1990, the business magazine, Forbes, has tracked the highest-earning athletes in the world. For each year, the magazine has listed the top athletes earning the most in salary and endorsements.

You are given a CSV data file, **athletes.csv**, which contains the information about the highest paid athletes from 1990 to 2021 listed by Forbes. You are asked to write program in the template **Q1_XXXXXXXX.java** to perform the following analysis on the data file.

(a) [10 marks] Implement the method loadCsvFile(). It loads the data from the CSV file and converts each row to an instance of Athlete (class provided in the template). This method should resolve all checked exceptions locally and return a list of athletes.

The CSV file has four fixed columns in this order: Name, Income, Year, Sport

(b) [10 marks] Implement the method highestIncome(). It finds the name of the athlete who has highest income in a given year. This method should be implemented using an imperative approach.

(c) [15 marks] Implement the method mostFrequentAthlete(). It finds the athlete who appears on the list for the greatest number of times. This method should be implemented using an imperative approach.

(d) [15 marks] Implement the method topTenIncome(). It finds the total income of the ten highest income athletes in a given year. This method should be implemented using a functional approach – that is to use Java Stream API and without external iterations.

(e) [15 marks] Implement the method mostFrequentSport(). It finds the sport which appears on the list for the greatest number of times. This method should be implemented using a functional approach – that is to use Java Stream API and without external iterations.

Question 2 (35 marks)

The program template Q2_XXXXXXXX.java is a generic task for the merge sort executed by fork-join framework. It can be used to sort any *Comparable* elements.

(a) [10 marks] Implement the generic method merge(). It merges two ascendingly sorted arrays into one. The merged array must maintain the ascending order. The time complexity of your implementation should be O(n).

(b) [15 marks] Implement the method compute(). It uses a divide-and-conquer algorithm that recursively divides the input into sorted sub-lists, typically single-element lists, and then merge them layer by layer.

(c) [10 marks] Implement the main() method to test the sorting. It uses a fork-join pool to execute this merge sort task (Q2_XXXXXXXX). You should generate an Integer array with five random numbers between 1 to 100 as the input for sorting. Print out the sorted array to validate your result accordingly.

- THE END -