

COP-3777 Programming II, Summer 2018

Programming Assignment 5: Advanced Data Structures in Java

Due Date: July 27 at 4:00 PM

In this assignment, you are asked to write classes *HashGradeBook* and *TreeGradeBook* that implement the following interface named *GradeBooking*.

```
public interface GradeBooking{
    void csvImport(String filename);
    void csvExport(String filename);
    void csvExport(String filename, Grade grade);
    void addGrade(String firstName, String lastName, int pID, String grade);
    Grade findGrade(String firstName, String lastName, int pID);
    void removeStudent(String firstName, String lastName, int pID);
    double findAverage();
};
```

Description of each method can be found below:

- *void csvImport(String filename)*: gets a file name as its only input parameter, reads its content and stores it in a map data structure (*java.util.HashMap<Student, Grade>* in *HashGradeBook* class and *java.util.TreeMap<Student, Grade>* in *TreeGradeBook* class). **The Java code of classes of *Student* and *Grade* are available on moodle.**

In the input csv file, every line contains the first name, last name, PID, and grade of a student separated by comma. Here is an example of input file format:

Alex,	Foster,	12345,	A-
Joe,	Miles,	98765,	B+
Amanda,	Taylor,	12398,	A

- *void csvExport(String filename)*: Transfers the students info from the map data structure back to the CSV file specified by its input parameter. The output file format is the same as the input file format used by the previous method.
- *void csvExport(String filename, Grade grade)*: Writes the information of students with the given grade to the CSV file specified by the first input parameter.

- *void addGrade(String firstName, String lastName, int pID, String grade)*: checks whether the student with the specification given by the input parameters currently exists in the Map data structure. If it does, the method changes his/her grade to the one given by the last input parameter; otherwise, it creates a key for the student and maps him/her to the grade given by the last input parameter.
- *Grade findGrade(String firstName, String lastName, int pID)*: returns the grade of the student whose information is given by the input parameters. If such student doesn't exist, it throws an exception.
- *void removeGrade(String firstName, String lastName, int pID)*: removes the student whose information is given by the input parameters from the map data structure. If such student doesn't exist, the method does nothing.
- *double findAverage()*: calculates and returns the average of all grades stored in the map data structure. The method throws an exception if the map is empty.

1 50% Extra Bonus Points

In the bonus part, you need to first write two classes *Hash1GradeBook* and *Hash2GradeBook* which both implement *GradeBooking* interface and store students information in hash maps *java.util.HashMap<Student1, Grade>* and *java.util.HashMap<Student2, Grade>* respectively (Implementation of classes *Student1* and *Student2* which are subclasses of class *Student* **is available on moodle**).

After implementing classes *Hash1GradeBook* and *Hash2GradeBook*, you are asked to write a program that does the following:

1. gets a CSV input file name from user through command line (the input file follows the format introduced before);
2. transfers the students information stored in the input file to the hash maps of both classes *Hash1GradeBook* and *Hash2GradeBook*;
3. determines which class uses a more efficient hash table for search operation. To do this, it needs to calculate the *average* running time of method *findGrade* when its input parameters get different values specified in the input file. To calculate the running time of a method, you can use *System.nanoTime()*.

Submissions

You need to submit a *.zip* file compressing the Java source file(s) of your program (*.java* files).