COP-3337 Programming II

Programming Assignment 1:
Designing Classes

Florida International University

1 Problem Specification

For this programming assignment, you will be completing the implementation of a java program for storing the grades of students in a *java.util.ArrayList* and calculating/presenting some statistics of grades.

To do this, you will complete the implementation of the following classes (the incomplete classes are available on Canvas and stored in "PA1 Draft.zip").

- *public class Grade*: keeps a single grade in the form of both score (from 0 to 100) and letter grade (A, A-, ..., F). The assumption is that the letter-grade cut-off points are the same as you see in the course syllabus.
- *public class Student*: keeps track of students info: first name (*String*), last name (*String*), 7-digit ID (*int*), and grade (*Grade*).
- *public class Gradebook*: keeps a list of students info in the following data structure: *java.util.ArrayList*⟨*Student*⟩.
- *public class Main*: contains the main method which does the followings: first it gets user's input data from *System.in* verifies the data to make sure there is no problem with the input data. Then, it asks for user's commands and gets them via *System.in*. Finally, it processes each command, and outputs the results to *System.out*.

Program Input Handling Phase

The program must first ask the user to enter the information of the first student via keyboard by printing the following message:

```
Welcome to my grade book!
Please enter the information of the first student using the following format:
"firstName lastName PID grade".
Press Enter when you are done.
```

After user enters the requested information (e.g. "Ann Smith 1234567 93"), the program must repetitively ask for information of the next student by printing the following message in a while loop:

> Please enter the information of the next student using the same format.
> If there is no more students, please enter the keyword "DONE".
> Press Enter when you are done.

The user input must meet the following criteria. If any of these criteria are not satisfied, the program must ask the user to try again:

- firstName is a single word which contains alphabetical characters only, starts with a capital letter and does not include any white space character.
- lastName is a single word which contains alphabetical characters only, starts with a capital letter and does not include any white space character. lastName may contain at-most one dot ('.') character too.
- PID is a seven digit integer with no leading zeros.
- grade is a non-negative integer that doesn't exceed 100.

Program Command Handling Phase

After the user enters the information of all students one-by-one and enters the keyword "DONE", your program must keep asking for a new command (e.g. print the message "Please enter a new command") and respond to each command properly. Here are the list of all commands that your program must support:

- *min score*: Your program must calculate and print the minimum score of all students as the response to this command.
- *min letter*: Your program must calculate and print the minimum letter-grade of all students as the response to this command.
- *max score*: Your program must calculate and print the maximum score of all students as the response to this command.
- *max letter*: Your program must calculate and print the maximum letter-grade of all students as the response to this command.
- *letter XXXXXXX*: Your program must find and print the letter-grade of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit).
- *name XXXXXXX*: Your program must find and print the full name of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit). Full name is made of first name, followed by a single white space and the last name.

- *change XXXXXXX YY*: Your program must find and update the grade of the student whose PID is given by the command (XXXXXXX is the PID where each X represents a digit). Int this command, YY is the symbol for the new grade.
- *average score*: Your program must calculate and print the average score (out of 100) of all students as the response to this command.
- *average letter*: Your program must calculate and print the letter-grade of average score of all students as the response to this command.
- *median score*: Your program must calculate and print the median score of all students as the response to this command.
- *median letter*: Your program must calculate and print the letter-grade of median score of all students as the response to this command.
- *tab scores*: Your program must print the list of all students in the form of a tab-separated table containing four columns with labels first name, last name, PID, and score.
- *tab letters*: Your program must print the list of all students in the form of a tab-separated table containing four columns with labels first name, last name, PID, and letter-grades.
- *quit*: Your program must stop asking for more commands and quit.

2 Submissions and Grading Criteria

You need to submit a zip file containing two packages "main" and "util" similar to the "PA1 Draft.zip" file given along with the assignment specification on Canvas. Here are the grading criteria:

- Code readability: 5%
- Using comments to explain every line of the program: 5%
- Correctness of Java syntax (no compilation error): 5%
- Proper usage and naming of variables and methods: 5%
- Proper implementation of Student, Grade, and Gradebook classes including their constructors: 40%
- Proper implementation of the main method including the input-handling phase (when the students' info are stored in an ArrayList⟨Student⟩) and the command-handling phase (when the program parses each command, calls appropriate methods, and prints the results): 40% (20% for input-handling and 20% for command-handling)