CSC 241 Assignment 6
Abstract Data Types and Programming Methodology
Due: Thursday, April 28

1 Introduction

In this semester, we are going to develop a Java program called "***GradeManager***", which manages students' grades. This program provides functions such as adding/deleting/editing grades of certain students. All the grades are stored in designated file(s) and updated as grading data change. It also supports to manage students, *e.g.*, adding a new student and his/her grading data or deleting a grading record of a particular student. This program will be built up through several assignments, in each of which you will be asked to apply what you will learn in lectures. At the end of semester, you will have a Java program which utilizes various OOP techniques and diverse data structures.

The assignments will be handed in an order for completing a final program. So, you MUST follow instructions and achieve requirements when you work on an assignment. Java code for each assignment should be errorless and submitted in the Blackboard course shell. Since a next assignment usually asks you to add more functions or edit what have been made in the prior assignment, you should keep the previous Java code(s). If the prior program is submitted with errors or runs unsuccessfully, it must be corrected before it goes to the next assignment.

2 Goal for This Assignment

In this assignment 6, you will implement and use stack for managing students' data. After your program finds data for students, the data is stored in a stack. From the addition of stack, you will learn how to use stack. Additionally, you will add a new function for calculating the final grade and the letter grade. The information can be used as a key for finding certain students. Through the new functions, ***GradeManager*** will be enhanced and increase usuability.

3 Instructions

A. Template file

Each assignment should be built in a package. The names of package and class for this assignment are below.

Package: Assignment6

Class: GradeManager

```
package Assignment6;
       …
public class GradeManager {
       …
```

Since there are no new files required for this assignment, no template package is uploaded. You MUST change the name for the package from Assignment5 to <u>Assignment 6</u> by yourself.

B. Data File / Properties File

Json file includes 20 students' grade data. If you successfully completed the assignment 5, you could add new students, or delete students.

From the assignment 3, `config.properties` file has been used to determine a full path for a json file. Once you set the file path correctly, you do not need to update any more.

C. Developing Environment

Your program should be **implemented in Java only**. The program in another language will not be graded.

D. Submission

You will submit your Java package. Zip the package **Assignment6** and upload it in the Blackboard course shell. DO NOT copy and paste your code into text files such as rtf, doc, or txt. You MUST <u>submit Java files</u>, not text files! The assignment will give you **two weeks** so it is **due on Thursday, April 28**. All submission **by 11:59 PM** on that day will be accepted without any penalty. On the due date, Blackboard may be suffering of too much network traffics and be unstable. There is no excuse about the issue, therefore you are strongly recommended to submit earlier than the due date.

4 Requirements

A. Complete all the requirements in the assignment 5

Regardless of new requirements for this assignment, your program should be able to do what were requested in the assignment 3 and 4. If you have already completed all requirements in the assignment 3 and 4, you would concentrate on the new requirements. Otherwise, you must complete them first because they are required for the assignment 5.

However, you do not need to do the benchmark for searching algorithms which was requested in the assignment 4. You will use only binary search for finding information.

B. Calculate final grade

In this assignment, you will finalize your grading work – calculating final grade and assigning letter grade. Final grade for each student must be calculated by a grading algorithm, which will be explained below. The letter grade is determined based on *final grade* (not the score of final exam). It is assigned by the chart shown in Table 1.

Table 1. Letter Grades

93 to up: $\boldsymbol{A}$	83 to 86.99: $\boldsymbol{B}$	73 to 76.99: $\boldsymbol{C}$	63 to 66.99: $\boldsymbol{D}$
90 to 92.99: $\boldsymbol{A}-$	80 to 82.99: $\boldsymbol{B}-$	70 to 72.99: $\boldsymbol{C}-$	60 to 62.99: $\boldsymbol{D}-$
87 to 89.99: $\boldsymbol{B}+$	77 to 79.99: $\boldsymbol{C}+$	67 to 69.99: $\boldsymbol{D}+$	Below 60: $\boldsymbol{E}$

Final grade is a weighted grade which is calculated by the following equation:

$$\text{Final Grade} = \sum \text{PERCENTILE COURSE WORK} \times \text{WEIGHT}$$

The Percentile course work is calculated by earned score over the maximum. For example, Daniel received 79 points. Weight for each course work is in Table 2.

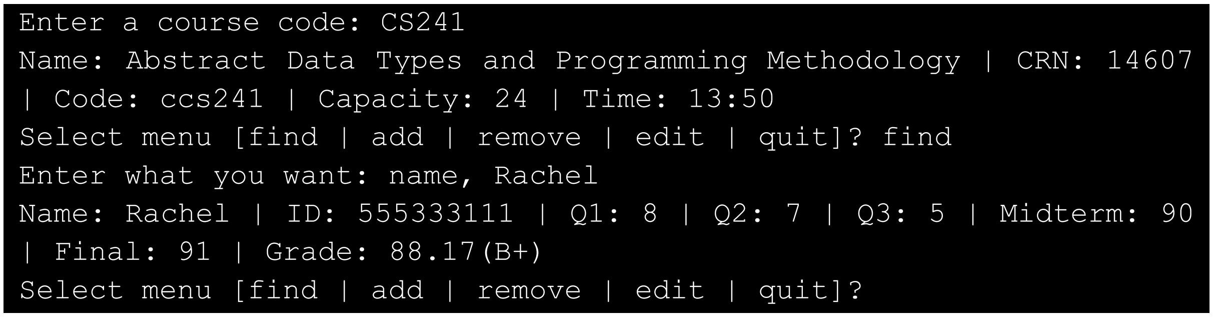
Table 2. Weight for Course Work

COURSE WORK	WEIGHT(%)
QUIZZES	10
MIDTERM	40
FINAL	50

Note that each quiz is equally weighted. In the assignment, all quizzes have the same maximum score that is 10, thus you can average those scores. For example, Daniel, who got 9, 5, and 7 for Q1, Q2, and Q3 respectively, will have 7 $(= (9 + 5 + 7)/30)$. If he received 79 and 57 for midterm and final respectively, his final grade is 67.1 which is calculated by,

$$\text{Final Grade} = \left(\frac{7}{10}\right) \times 10 + \left(\frac{79}{100}\right) \times 40 + \left(\frac{57}{100}\right) \times 50$$

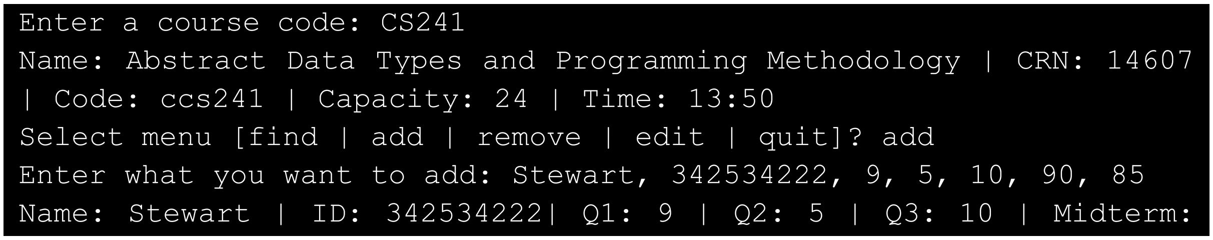
According to Table 1, his final grade must be $\boldsymbol{D}+$. The final grade and the letter grade of a student should be calculated when you request to find the student's data. For example, when you ask to find a student Rachel, you should see her final grade which is 88.17 and $\boldsymbol{B}+$ shown in Figure 1.



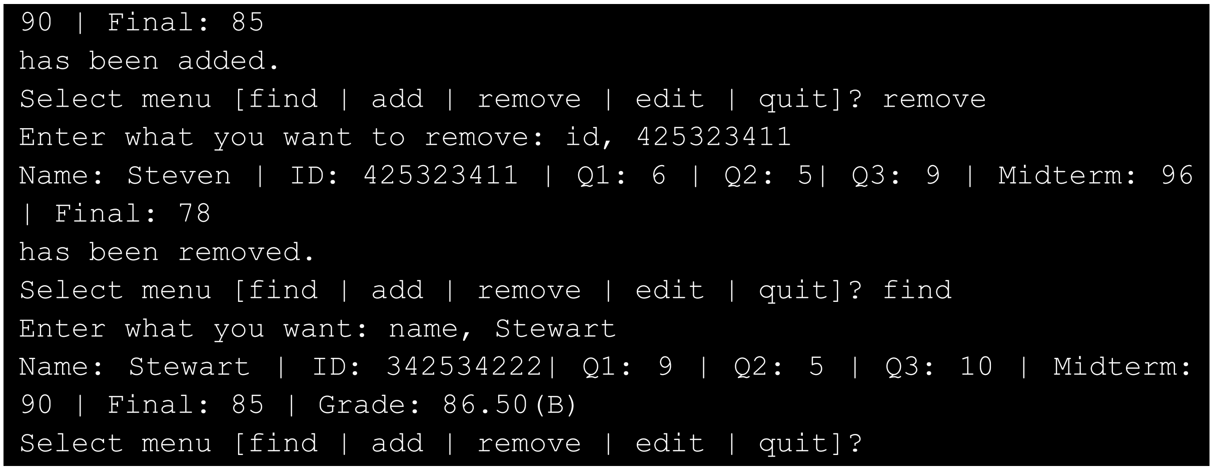
```
Enter a course code: CS241
Name: Abstract Data Types and Programming Methodology | CRN: 14607
| Code: ccs241 | Capacity: 24 | Time: 13:50
Select menu [find | add | remove | edit | quit]? find
Enter what you want: name, Rachel
Name: Rachel | ID: 555333111 | Q1: 8 | Q2: 7 | Q3: 5 | Midterm: 90
| Final: 91 | Grade: 88.17(B+)
Select menu [find | add | remove | edit | quit]?
```

Figure 1. After your program find a student's data, it should calculate the final grade and show it with the letter grade.

However, Grade does not have to be shown when a student is added or deleted. As shown Figure 2, when your program is requested to "add" and "remove" a student, it does not print the final grade and the letter grade for the student. Similarly, when a student is edited, the final grade and the letter grade does not have to be calculated.



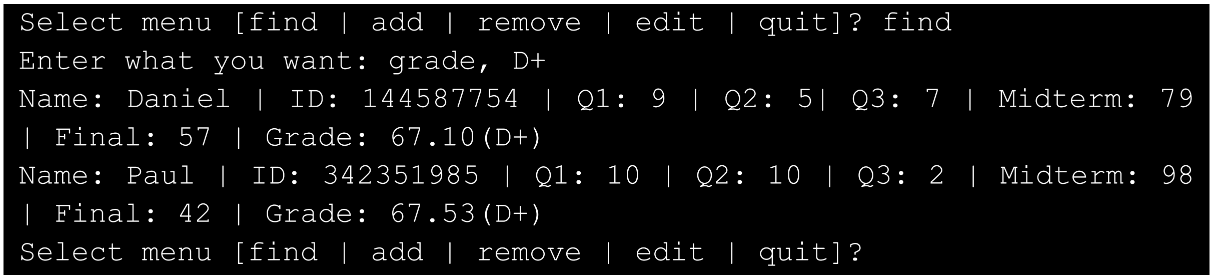
```
Enter a course code: CS241
Name: Abstract Data Types and Programming Methodology | CRN: 14607
| Code: ccs241 | Capacity: 24 | Time: 13:50
Select menu [find | add | remove | edit | quit]? add
Enter what you want to add: Stewart, 342534222, 9, 5, 10, 90, 85
Name: Stewart | ID: 342534222| Q1: 9 | Q2: 5 | Q3: 10 | Midterm:
```



```
90 | Final: 85
has been added.
Select menu [find | add | remove | edit | quit]? remove
Enter what you want to remove: id, 425323411
Name: Steven | ID: 425323411 | Q1: 6 | Q2: 5| Q3: 9 | Midterm: 96
| Final: 78
has been removed.
Select menu [find | add | remove | edit | quit]? find
Enter what you want: name, Stewart
Name: Stewart | ID: 342534222| Q1: 9 | Q2: 5 | Q3: 10 | Midterm:
90 | Final: 85 | Grade: 86.50(B)
Select menu [find | add | remove | edit | quit]?
```

Figure 2. When a student's data is added or deleted, his/her final grade and letter grade are not shown. The information can be found when it is requested to find.

Since Grade is newly added as information for student, it can be used for a key to find certain students. The grade could be final grade which is numeric or letter grade which is alphabet. Figure 3 illustrates that your program finds students whose grade is $\boldsymbol{D} +$: Daniel and Paul.



```
Select menu [find | add | remove | edit | quit]? find
Enter what you want: grade, D+
Name: Daniel | ID: 144587754 | Q1: 9 | Q2: 5| Q3: 7 | Midterm: 79
| Final: 57 | Grade: 67.10(D+)
Name: Paul | ID: 342351985 | Q1: 10 | Q2: 10 | Q3: 2 | Midterm: 98
| Final: 42 | Grade: 67.53(D+)
Select menu [find | add | remove | edit | quit]?
```

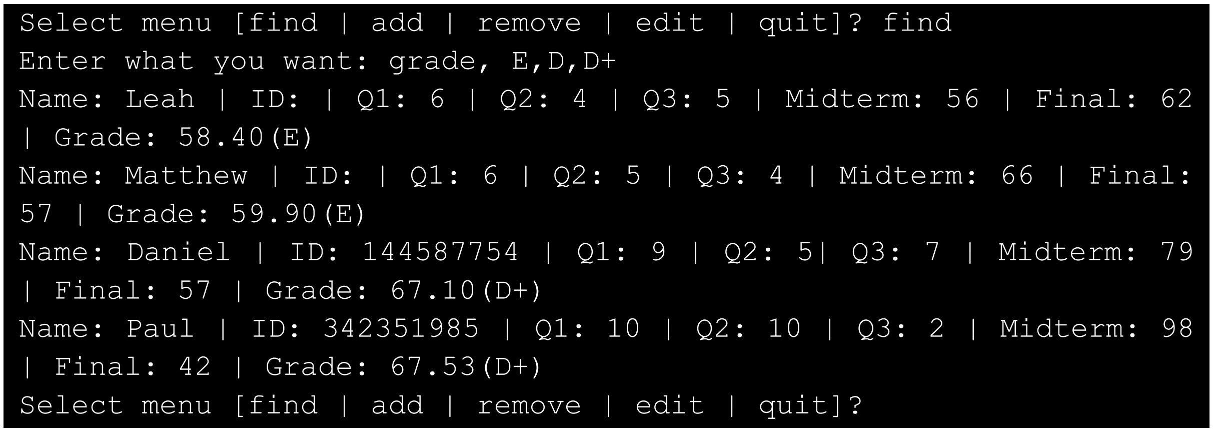
Figure 3. Students whose grade is under a given condition are found and displayed. Remember that their final grades and letter grade should be shown.

C. Stack for found students

When your program finds students by a particular condition upon request, it should have data for the students in a stack structure which was learned in Lecture 11. Note that data must be pushed into stack regardless of the number of students. There is one student found in Figure 1 and Figure 2, while there are two students in Figure 3. Either case, your program MUST keep data in a stack.

D. [Extra Work] Enhanced conditional search

It is an extended version from extra work in the assignment 5, which allowed to have a ranged condition, such as "final, 91-96". You will add more complex condition which has "," (comma). For instance, you may want to find students whose grade is $\boldsymbol{E}$, $\boldsymbol{D}$ or $\boldsymbol{D} +$, shown in Figure 4: Daniel, Leah, Matthew, and Paul. As instructed in the section C, the data of the students must be stored in a particular data structure – stack. In the extra work, however, you will use a different dynamic data structure which is priority stack which is introduced in Lecture 11. Priority queue benefits handling data in a particular order, generally natural order. Therefore, the students under the condition are printed in order in Figure 4: Leah, Matthew, Daniel, and Paul are ordered by the final grade.



```
Select menu [find | add | remove | edit | quit]? find
Enter what you want: grade, E,D,D+
Name: Leah | ID: | Q1: 6 | Q2: 4 | Q3: 5 | Midterm: 56 | Final: 62
| Grade: 58.40(E)
Name: Matthew | ID: | Q1: 6 | Q2: 5 | Q3: 4 | Midterm: 66 | Final:
57 | Grade: 59.90(E)
Name: Daniel | ID: 144587754 | Q1: 9 | Q2: 5| Q3: 7 | Midterm: 79
| Final: 57 | Grade: 67.10(D+)
Name: Paul | ID: 342351985 | Q1: 10 | Q2: 10 | Q3: 2 | Midterm: 98
| Final: 42 | Grade: 67.53(D+)
Select menu [find | add | remove | edit | quit]?
```

Figure 4. With the complex condition, your program may find multiple students. Note that the students are displayed in order.

It is not mandatory. Those who complete it, will have extra credits up to 20% (which is worth 6 points).

5 Grading

A. Grading criteria

The lab is assigned **30** points, which is 10% of the final grade. It will be graded by evaluating the requirement. Any missing and unsatisfiable criteria will take off points. The tentative and brief criteria are below.

Compilation:	**5** points
Execution:	**5** points
Proper output:	**20** points
Extra work:	**6** points

B. Late penalty

Late submission will take off **2 points per day** after due date. **Thus, submission after 15 days will not be accepted in any circumstances**.

6 Academic Integrity

Any dishonest behaviors will not be tolerated in this class. Any form of plagiarism and cheating will be dealt with according to the guidelines on the Academic Integrity Policy online at http://www.oswego.edu/integrity. For more information about university policies, see the following online catalog at:

http://catalog.oswego.edu/content.php?catoid=2&navoid=47#stat_inte_inte

Student who is against the honor code will not have any credits in this project.