CSC 241 Assignment 7
Abstract Data Types and Programming Methodology
Due: Friday, May 13

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

Through this assignment 7, you will have an opportunity to implement a class for Binary Search Tree (BST), which will be used to print data of students under a given condition. Similar to the *find* menu, the *print* menu needs to find desired students first and then print the data in a text file. The difference from *find* menu is that *print* menu stores students' data in a BST. Since the data are maintained in a tree format, a particular way must be used when it print them. Review lecture 12 for further information first, and then read the following sections carefully.

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: Assignment7

Class: GradeManager

```
package Assignment7;
        …
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 Assignment6 to Assignment 7 by yourself.

B. Data File / Properties File / Output 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.

Print menu, which is newly added in this assignment, generates an output file which includes data of students in order. The output file name is "`course_code-output.txt`". For example, if you chose "cs241" course, then the output file name is "`cs241-output.txt`".

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 **Assignment7** 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 submit Java files, not text files! The assignment will give you **two weeks (+ *a*)**, so it is **due on Friday, May 13**. 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. Methods for calculating final grade from requirements from assignment 6

Since conditions to print data of students could be anything such as q1, q2, or grade, methods for calculating final grade, which was one of the requirements for assignment 6, should be precedently done. If you had not complete them yet, you need to finish it before you start new requirements for the assignment 7.

B. Print data of students in a given condition

Through this assignment, the program ***GradeManager*** provides one more menu, which prints data of students under a given condition shown in Figure 1. A ranged condition must be allowed in this assignment, for instance, "`grade, 0-69.99`." Recall that data of students found by *find* menu were pushed into a stack. The data did not have to be in order in a stack, but they could be sorted by using priority queue. BST can make a similar output

as priority queue; first it constructs a BST with data. Then it prints all data through ***inorder*** traversal.

```
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 | print | quit]? print
What students will be printed: grade, 0-69.99
Name: Leah | ID: 345678232 | Q1: 6 | Q2: 4 | Q3: 5 | Midterm: 56 |
Final: 62 | Grade: 58.40(E)
Name: Matthew | ID: 246864248 | Q1: 6 | Q2: 5 | Q3: 4 | Midterm:
66 | Final: 57 | Grade: 59.90(E)
Name: Daniel | ID: 144586754 | 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 | print | quit]?
```

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

As shown in the slide 21 in lecture $12a$, when all the tree nodes are traversed by ***inorder*** strategy, nodes will be accessed in this order; left child node – root (or parent node) – right child node. Note that the tree nodes are recursively visited. Since left child node is less than parent node and parent node is less than right child node, this traversal strategy finally reads all data in order. For more information, you may review the lecture and slide.

C. [Extra Work] Enhanced Binary Search Trees

When a key datum is inserted in a BST, a proper spot must be found first by checking the condition, that is, is the key is less than, or equal/greater than datum in a current node. When the key which is identical to any of existing data is inserted, it will be placed somewhere in subtree by the rules. Consider a BST constructed by the input with 80, 85, 74, 80, 79, 65, 80, which is illustrated in **Figure 2**. By the condition, duplicates of 80 will be placed on the right subtree of the root node. It raises a problem when 80 should be found, because the BST always return 80 in the root.

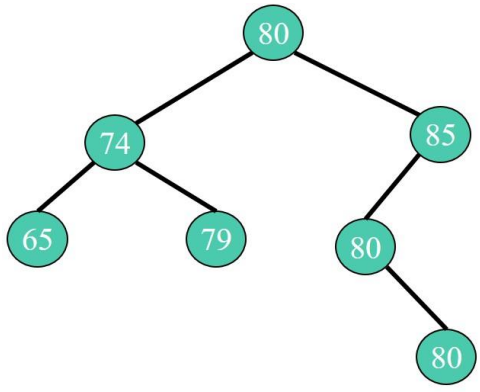

Figure 2. Binary Search Tree with duplicates.

In this extra work, you will improve BST, which solve the problem. Since it does not happen when all data are different, you may make BST with distinct data. Thus, one of the solutions is to make distinct nodes only. For duplicates data, you may put them in an existing node by using chaining. You will think of this problem, find(or make) a solution.

It is not mandatory. Those who complete it, will have extra credits up to 20% (which is worth 6 points). If you do this extra work, leave a short message in comment.

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

Since it is the last assignment, late submission will not be accepted. You have to submit by **Friday, May 13**.

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.