**CS 2050 - Programming Project # 4**

Assign Date: Session 14.

Due Date: Session 19. This project cannot be resubmitted.

**Auxiliary Documents: 2050 Project Estimating and Scheduling.docx** Complete initial and final schedules.

**Problem Scenario**

Metropolitan State University of Denver needs a Java programmer to assist them with creating a grade book application. You have graciously volunteered to help. The university wants to track their students and their grades in various courses. For each student, the university wants to track this information:

1. Student id (String – should be a unique value)
2. Student first name (String)
3. Student last name (String)
4. Student email address (String – should be a unique value)

The university wants to make sure that they have a correct email address for each student. They want you to verify there is a “@” in the email address.

For each grade item, the university wants to track this information:

1. Student id (String – should match a student in the student list)
2. Grade Item id (Integer – should be a unique value)
3. Course id (String)
4. Item type (String – must be one of the following: HW, Quiz, Class Work, Test, Final)
5. Date (String – format yyyymmdd)
6. Maximum score (Integer – must be greater than 0)
7. Actual score (Integer – must be in the range of 0 to maximum score)

**Note:** All your code will be in the main class and in the List class (described below). Do not make any changes to the Student and Grade Item classes created for Programming Project # 1. Your main class will have the name: FirstnameLastName\_04 (Your first and last name)

**Program Requirements**

In programming Project # 1 we created the Student class and Grade Item class. In this programming project, **we will program the List class using Linked List as discussed in class** (Chapter 12). The List class will be used to create a list of Students and Grade Items and **will implement MyCollectionInterfaceProject04**. Write pseudocode before writing any code. You will submit your pseudocode with the project.

Your main method will create two linked lists using the List class: one for a list of Students and another for a list of Grade Items. The main method will not know the data structure implemented by the List class.

Input files for this program are called YourName\_04\_Inputxx.txt, xx = 01, 02, 03, … The file contains lines of data about Students and Grade Items. An action field is added to each line as the second field which indicates whether to add or remove the specified student or grade item.

The format of an input file is as follows and will contain multiple lines of data.

STUDENT,ADD,studentId,firstName,lastName,emailAddress

STUDENT,DEL,studentId,firstName,lastName,emailAddress

GRADE ITEM,ADD,id,studentId,courseId,type,date,maxScore,actualScore

GRADE ITEM,DEL,id,studentId,courseId,type,date,maxScore,actualScore

**Sample Input File**

STUDENT,ADD,900123456,Joe,Doe,joedoe#msudenver.edu

STUDENT,ADD,900123456,Joe,Doe,joedoe@msudenver.edu

STUDENT,ADD,900123456,Joe,Doe,joedoe@msudenver.edu

GRADE ITEM,ADD,1,900123456,23456,HW,20190112,100,95

STUDENT,ADD,900123457,Jane,Doe,janedoe@msudenver.edu

STUDENT,ADD,900123458,Mary,Smith,marysmith@msudenver.edu

STUDENT,ADD,900123459,David,Smith,davidsmith@msudenver.edu

GRADE ITEM,ADD,2,900123456,23456,HW,20190113,100,75

GRADE ITEM,ADD,3,900123456,23456,HW,20190114,100,95

STUDENT,ADD,900123457,Joe,Doe,joedoe@msudenver.edu

STUDENT,DEL,900123456,Joe,Doe,joedoe@msudenver.edu

GRADE ITEM,DEL,1,900123456,23456,HW,20190112,100,95

STUDENT,DEL,900123456,Joe,Doe,joedoe@msudenver.edu

The input files should be placed in the same folder as your main class. You can refer to these files using the following String variable.

final String INPUT\_FILE = "YourName\_04\_Inputxx.txt";

where xx is the input file number 01, 02, 03, ... Create multiple input files to test various cases.

**The Main class**

The UML diagram for your main class is given below. All fields and methods are static. Follow the Java Programming Style Guide to name your classes, methods and variables:

|  |
| --- |
| Main Class |
| - listOfStudents : List  - listOfGradeItems : List  - INPUT\_FILE : String  - OUTPUT\_FILE : String |
| + main (args : String[]) : void  + processInput () : void  + processStudentData (info : String[] ) : void  + processGradeItemData (info : String[] ) : void  + generateReport () : void |

The main method will:

1. Instantiate the listOfStudents and listOfGradeItems objects
2. Call the processInput method
3. Call the generateReport method

The processInput method will:

1. Open the input file – display an error message using err stream if the file is not found and exit.
2. Read the data line.
3. Use the split() method to parse the line
4. If the first field is STUDENT, call the processStudentData method passing it the array of type String containing the split Student data. If the first field is GRADE ITEM, call the processGradeItemData method passing it the array of type String containing the split GradeItem data. If the first field is neither STUDENT or GRADE ITEM, display an error message with the line and exit the processInput method (not the program).

The processStudentData method will keep processing unless it encounters and error, in which case the method will display a message using the err stream and exit the processStudentData method. The method’s processing is as follows:

1. Receive an array of type String with one line of Student data and create a Student object from the input data to use with subsequent steps.
2. If the second field is ADD:
   1. Use the contains method in the ist class to make sure Student is unique (not in the list yet). If Student is already in the list then display an error message using err stream.
   2. Call the add method in the List class to add the Student object to the list of students.
   3. Check the return value from the add method and if it failed, display a message using err stream.
3. If the second field is DEL:
   1. Call the remove method to remove the Student from the list of students.
   2. Check the return value from the call to remove method and if it is null, display a message using the err stream.
4. If the second field is neither ADD nor DEL, display an error message and continue with the next record.
5. The processStudentData method must catch any exception thrown by the constructor in the Student class and display the accompanying error message using the err stream.
6. The processGradeItemData method will:
7. Receive an array of type String with GradeItem data and create a GradeItem object from the input data to use with subsequent steps.
8. If the second field is ADD:
   1. Use the contains method in the List class to make sure GradeItem is unique (there is no other entry that contains the same student id and grade item number). If GradeItem is already in the list, display an error message using err stream.
   2. Call the add method in the List class to add the GradeItem object to the list of grade items.
   3. Check the return value from the add method and if it failed, display a message using the err stream.
9. If the second field is a DEL then it will:
   1. Call the remove method to remove the GradeItem from the list of Grade Items.
   2. Check the return value from the call to remove method and if it is null, display a message using the err stream.
10. If the second field is neither ADD nor DEL, display an error message and continue with the next record.
11. The processGradeItemData method must catch any exception thrown by the constructor in the GradeItem class and display the accompanying error message using the err stream.

The generateReport method will:

1. Call the toArray method in the List class to get a list of Student objects.
2. Call the toArray method in the List class to get a list of GradeItem objects.
3. Generate a report which will be written to an output file, Project4outputxx.txt, xx = 01, 02, 03, … (and corresponds to Project4inputxx.txt) which will be located in the same folder as your main class. The output file names should correspond to the input file names, like Project4input01.txt should produce Projectoutput01.txt, and so on. **Output files should include console dialog.**

**Output Report Format**

For each Student object in the Student list, the code will examine each Grade Item in the Grade Item list and find the Grade Item with Student IDs that match the Student ID of the Student object. It will display the information using the following format:

StudentID FirstName LastName Email

Grade Items

Maximum Actual

GradeItemID CourseID Type Date Score Score Grade(%)

===================================================================================

Total SumOf SumOf

MaxScore Actual Grade(%)

Grand Total

Sample Output Report

900123456 Joe Doe joedoe@msudenver.edu

Grade Items

1 12345 HW 20190112 100 95 95.0%

2 12345 Class Work 20190113 100 75 75.0%

3 12345 Test 20190217 500 345 69.0%

============================================================

Total 700 515 73.6%

900123457 David Brown david@gmail.com

Grade Items

1 12346 HW 20190112 100 95 95.0%

2 12346 Class Work 20190113 100 75 75.0%

3 12346 Test 20190217 500 340 68.0%

============================================================

Total 700 510 73.6%

Grand Total 1400 1025 73.2%

Sample Display Output (displayed on screen while processing the Sample Input File shown on page 2). The xxxxxxx below refers to the relevant student ids.

Reading data from file Project4Input01.txt

Error: Email address joedoe#msudenver.edu is invalid  
Student was not added to the students list.

Student with Student Id 900123456 was added to the list.

Student with Student Id 900123456 is already in the list.

Grade Item with Student id xxxxxxx and Grade Item Id 1 was added to the list.

Student with Student Id 900123457 was added to the list.

Student with Student Id 900123458 was added to the list.

Student with Student Id 900123459 was added to the list.

Grade Item with Student id xxxxxxx and Grade Item Id 2 was added to the list.

Grade Item with Student id xxxxxxx and Grade Item Id 3 was added to the list.

Student with Student Id 900123457 was added to the list.

Student with Student Id 900123456 was removed from the list.

Grade Item with Student id xxxxxxx and Grade Item Id 1 was removed from the list.

Student with Student Id 900123456 is not in the list.

Generating report ... done.

**What to Submit**

Your final output schedule on top of your main class (FirstnameLastname\_04.java), on top of your pseudocode, on top of your LinkedList class, on top of pairs of input and output files, all stapled together. The order of the input/output files should be Project4input01.txt, Project4output01.txt, Project4input02.txt, Project4output02.txt, etc. Write your name and Project 4 in the upper right-hand corner of the top page.

***Submit your Student and/or GradeItem class(es) only if you made changes to it/them***, in which case submit it/them on top of the input and output files. Highlight any changes in the Student and/or GradeItem classes with a comment line consisting of:

// \*\*\*\*\* CHANGE CHANGE CHANGE CHANGE CHANGE CHANGE CHANGE CHANGE \*\*\*\*\*

You will lose points if you do not follow these submission instructions.

Note: always submit source code with line numbers.