**CS 2050 - Programming Project # 3**

Assign Date: Session 7.

Due Date: Session 13. Can be resubmitted ONE TIME Session 17 if your first submission is in by the first due date.

**Problem Scenario**

Metropolitan State University of Denver needs a Java programmer to assist it 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 in this project is in the main class named

FirstnameLastName\_03 (Use YOUR first and last name!)

and in the List class you write as specified below. Do not make any changes to the Student and Grade Item classes created for Programming Project # 1 unless you still have to make corrections to it.

**Program Requirements**

In Project # 1, we created the Student class and Grade Item class. In this project, we implement the List class (container) which can be used to create a list of Students and Grade Items.

The List class implements the MyCollectionInterface. Your main method creates two lists using the List class: one for list of Students and another for list of Grade Items.

The input files for this program are called “hw3inputxx.txt”, where xx = 01, 02, 03, … Each file contains lines of data about Students and Grade Items. An action field is added to each line as the second field, to indicate whether to add or remove the specified student or grade item. (See the Addendum in the Project 2 specification for tips on easily changing the xx.) Output tests are also separated by a blank line, a row of \*s and another blank line.

The format of the input file is as follows and contains 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 01**

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

This file should be placed in the same folder as your main class. You can open this file using the following String variable.

final String INPUT\_FILE = "hw3inputxx.txt";

where xx is the input file number, xx = 01, 02, 03, …. You can create multiple input files to test the cases required in this project.

Your main class does not know the data structure implemented by your List class. We use the same main class with future programming projects with minimal changes. The List class will be modified in a future project to reflect a different data structure.

**The Main class**

The UML diagram for your main class is given below (all fields and methods are static). Please follow the Java Programming Style Guide (file 56 in OneDrive > Foundation) to name your classes and methods:

|  |
| --- |
| 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:

1. Instantiates the listOfStudents and listOfGradeItems object
2. Calls the processInput method
3. Calls the generateReport method

**Note**: in all cases below, if an error is detected, display a message with useful information to an end‑user using the err stream and move on to the next test.

The processInput method:

1. Opens the input file – display an error message using the err stream if the file is not found. Include the name of the missing file in the message (Be end-user friendly)
2. Reads the data line
3. Uses split to parse the line
4. If the first field is STUDENT, calls the processStudentData method passing it the array created by the split method of type String containing Student data. If the first field is GRADE ITEM, calls the processGradeItemData method passing it the array created by the split method of type String containing Grade Item data.

The processStudentData method:

1. Receives an array of type String with Student data.
2. If the second field is ADD then it:
3. Creates a Student object.
4. Uses the contains method in the List class to make sure Student is unique. If Student is already in the list, display an error message that includes the student ID.
5. Call the add method in the List class to add the Student object to the list of students.
6. If the add method’s return value indicates an error, display a message with the Student ID.
7. If the second field is a DEL then it:
   1. Creates a Student object.
   2. Calls the remove method to remove the Student from the list of students.
   3. Checks the return value from the call to the remove method. If the entry was not found, display a message using the err stream.
8. processStudentData method must catch the exception thrown by the constructor in the Student class and display an error message usingthe err stream.
9. If the second field is neither ADD or DEL, write an error message saying so, showing the second field.

The processGradeItemData method:

1. Receives an array of type String with GradeItem data
2. If the second field is ADD then it:
3. Creates a GradeItem object.
4. Uses the contains method in the List class to make sure Grade Item is unique. If the Grade Item is already in the list, display an error message that includes the Student ID and Grade Item ID.
5. Calls the add method in the List class to add the Grade Item object to the list of grade items.
6. If the add method’s return value indicates and error, display a message that includes the Student ID and Grade Item ID.
7. If the second field is a DEL then it:
   1. Creates a GradeItem object.
   2. Calls the remove method to remove the GradeItem from the list of Grade Items.
   3. Checks the return value from the call to remove method. If the item was not found, display a message using the err stream.
8. The processGradeItemData method must catch the exception thrown by the constructor in the GradeItem class and display an error message using the err stream.
9. If the second field is neither ADD or DEL, write an error message saying so, showing the second field.

The generateReport method:

1. Calls the toArray method in the List class to get a list of Student objects.
2. Calls the toArray method in the List class to get a list of GradeItem objects.
3. Generates a report which is written to an output file, hw3outputxx.txt, xx = 01, 02, 03, …, located in the same folder as your main class. Output file hw3outputxx.txt corresponds to input file hw3inputxx.txt.

**Output Report Format**

For each Student object in the Student list, the code examines 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 displays the information using the following format:

StudentID FirstName LastName Email

Grade Items

GradeItemID CourseID Type Date Maximum Score Actual Score Grade (%)\*

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

Total SumOfMaxScore SumOfActualScore Grade (%)

Sample Output Report

900123456 Joe Doe joedoe@msudenver.edu

Grade Items

1 12345 HW 20190112 100 95

2 12345 Class Work 20190113 100 75

3 12345 Test 20190217 500 345

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

Total 700 515 73.6%

900123457 David Brown david@gmail.com

Grade Items

1 12346 HW 20190112 100 95

2 12346 Class Work 20190113 100 75

3 12346 Test 20190217 500 345

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

Total 700 515 73.6%

\*The “Grade (%)” is the sum of the actual score / the sum of the maximum score \* 100.

Sample Display Output (displayed on screen while processing the input file):

Reading data from file hw3Input01.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 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 Grade Item Id 2 was added to the list.

Grade Item with 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 Grade Item Id 1 was removed from the list.

Student with Student Id 900123456 is not in the list.

Generating report to file hw3output01.txt... done.

**What to Submit**

Your main class, FirstnameLastname\_03 (use YOUR name!), on top of your Link class, on top of pairs of input and output files, all stapled together. The order of the input/output files should be hw3input01.txt, hw3output.txt01, hw3input02.txt, hw3output02.txt, etc. Write your name and Project 3 in the upper right-hand corner of the top page. Note you do not have to submit your Student and GradeItem classes as they should be unchanged from Project 02. If you did make changes to these two classes, indicate in pen or highlighter what you want graded. Resubmitting? Place the earlier submission at the end and indicate on the top page of the new submission that you are resubmitting. Reminder: Project 03 can be resubmitted only once.