**Project 4 Expectations**

Please review ***General Programming Expectations*** document located in Week 2 under **Week 2 Checklist**.

**Obtaining data for Assignment 4**

The assignment does not include any student data or other information about the students. For consistency in conversation and grading of the programs, there is a data file, ***StudentData.txt***, that is attached to this assignment. The file can be read into the program and assigned to the appropriate HashMap or the data can be initialized into the HashMap. The data contained in the ***StudentData.txt*** file:

7623:Phyllis Jones,English,0

8729:Cletus Smith,History,0

7321:Betty Booth,Computer Science,0

3242:Samuel Seybright,History,0

9823:Oscar Blu,English,0

2341:Sally Grief,Computer Science,0

8321:Jacques Matchel,Mathematics,0

The description of the Student HashMap is

Key Student ID

ArrayList Student First Name

Student Last Name

Major

GPA

**Note**: There is a difference in the description from the elements of the database detailed in the .pdf assignment document. There are two reasons for this change: (1) the GPA is a value that is expected to be displayed as part of the Find process and (2) including the value when the program begins processing allows for a comparison between the starting HashMap and the final HashMap. [This requirement is an additional expectation of the assignment.]

**Suggested Design for Assignment 4**

Here is one skeleton design of this program.

package cmis242.prj4smithx;

[imports here]

public class CMIS242 PRJ4SmithX {

public static class GUI {

*[Constructor (), read in file and assign to HashMap, initialize HashMap]*

*[Design the GUI window]*

*[Invoke listener]*

} // end public static class GUI

public static void WriteHashMap {

[Display all elements of the HashMap in the IDE Output Window]

} // end public static void WriteHashMap

public static class Student {

[Declare Student information, method to generate GPA, toString]

} // public static class Sequence

public static void main (String args[]) {

*[Declare instance of GUi class]*

} // end public static void main (String args []) {

} // end public class CMIS242 PRJ4SmithX

**There are some considerations in designing the program**:

* ***Exit/End*** button should be added to the window below the ***Process Requested*** button.
* When the GUI window first displays the program, write to the *IDE Output Window* all the elements of the database each on its own line. This is one possible format of that output:

7623 Phyllis Jones English 0

* When the GUI window first displays only the ***Student ID*** TextField should be accessible.
* The remaining two TextFields (Name and Major) of the GUI window should only be accessible and contain text when the ***Insert*** operation is selected.
* For every entry of a value in the Student ID TextField, there is a check to determine if the ID is part of the HashMap database. If the entry is not in the HashMap database, a JOption Pane titled *No Record* is displayed with a message that the number entered is not in the database.
* Under normal circumstances, there would be processing to attempt adding the missing record to the database, but that is not part of the requirements for this program.
* When the ***Find*** operation is selected, the JOption Pane titled *Locate Student* opens with the following values including labels displayed on separate lines: (1) the student ID, (2) the first and last name, (3) the major, and (4) the GPA. There is an OK button at the bottom of the pane that returns to the initial GUI window.
* When the ***Delete*** operation is selected, the JOption Pane titled *Delete Record* opens and displays the Student ID and a message about the deletion from the database.
* When the ***Update*** operation is selected, a JOption Pane titled *Grade Change* displays the Current GPA and then using .showInputDialog () lists the grades. Once a grade is selected, either the ***OK*** or ***Cancel*** button is clicked. [For this assignment there is only the requirement to add a single course to the database. Under normal circumstances, it would be expected that many courses be added to the database. That is not part of this assignment. Processing one GPA is enough.]
* With a grade selected and the OK button clicked, a JOption Pane titled *Credits* displays and then using .showInputDialog () lists the possible credits (1-6). Once a credit is selected, either the ***OK*** or ***Cancel*** button is clicked.
* When the ***OK*** button of the *Credits* window is clicked, the GPA is calculated and assigned to the HashMap GPA element.
* Once the GPA processing is complete, the JOption Pane used for the ***Find*** operation is displayed with the new information.
* After processing several times based on the Test Plan scenarios, the user clicks the ***Exit/End*** button.
* At the close of the program – clicking ***Exit/End*** – the HashMap is again displayed in the *IDE Output Window* with the same format and the modified results from the various operations when the program ran.

7623 Phyllis Jones English 3

* Review the two displays in the IDE Output Window for the differences resulting from the processing.