University of Maryland University College

CMIS 242 6382 Intermediate Programming (2165)

Submit Files - Project 4

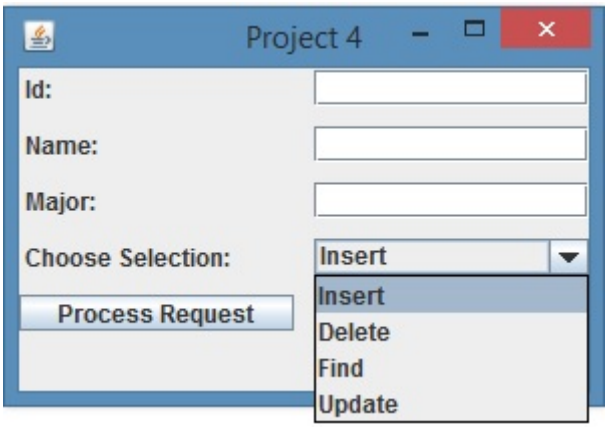
Hide Submission Folder Information

Submission Folder

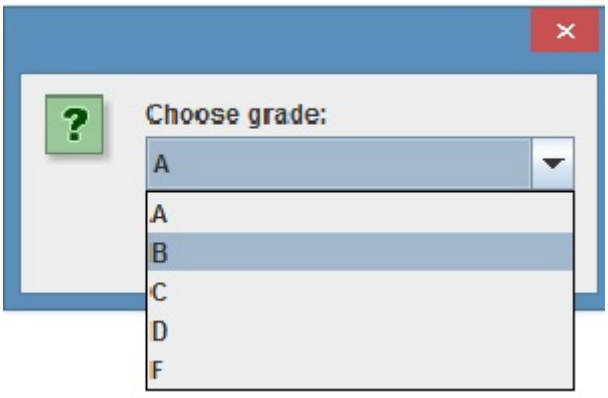
Project 4

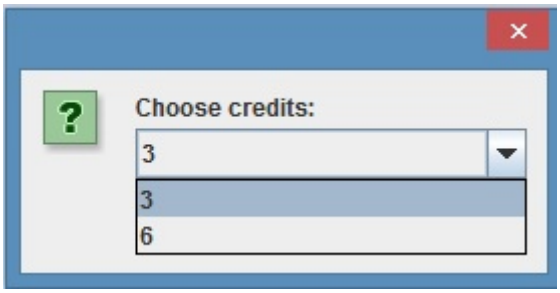
CMIS 242 6382 Intermed... Justin Casteel

The fourth programming project involves writing a program to manage a student database. The interface to the program should be a GUI that looks similar to the following:



A combo box should allow the user to select one of the four database actions shown. The database should be implemented as a `HashMap`, with the ID field as the key and a student record consisting of a name and major as the value. The operation should be performed when the user clicks the *Process Request* button. If the user attempts to insert a key that is already in the database an error message should be displayed using a `JOptionPane` message dialog box. If the user attempts to delete, find or update a record that is not in the database, a message should also be displayed. After each successful operation is completed a `JOptionPane` window should be displayed confirming the success. In the case of a successful *Find* request, a window should pop up containing the student's ID, name, major and current GPA. When the user selects the *Update* request, the following `JOptionPane` windows should be displayed to gather information about a course that has just been completed:





This program must consist of two classes. The first class should define the GUI and handle the database interactions. It should be hand-coded and not generated by a GUI generator. The second class named `Student`, should define the student record. It must have instance variables for the student name, major and two variables that are used to compute the GPA. A variable that contains the total number of credits completed and a second variable that contains the total quality points, which are the numeric value of the grade received in a course times the number of credit hours. It should not contain the student ID. The class should have the following three methods:

1. A constructor that is used when new student records are created. It should accept the name and major as parameters and initialize the fields that are used to compute the GPA to zero.
2. The second method `courseCompleted` should accept the course grade and credit hours and update the variables used to compute the GPA. It will be called when an *Update* request is made.
3. The third method should override `toString` and return a labeled string containing the student name, major and GPA.

Be sure that all instance and class variables are declared as *private*. Also any exceptions thrown by nonnumeric inputs should be properly handled. Finally when a student has not yet completed any course, the GPA should be displayed as 4.0.

Submit Cancel

CMIS 242 6382 Intermed... Justin Casteel

Hide Rubrics

Rubric Name: Assignment Rubric

Criteria	Exceeds	Meets	Does not meet
Design	20 points (18-20 points) Employs Modularity (including proper use of parameters, use of local variables etc.) most of the time Employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) most of the time Efficient algorithms used most of the time Excellent use of object-oriented design	17 points (15-17 points) Employs Modularity (including proper use of parameters, use of local variables etc.) some of the time Employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) some of the time Efficient algorithms used some of the time Good use of object-oriented design	14 points (0-14 points) Rarely employs Modularity (including proper use of parameters, use of local variables etc.) Rarely employs correct & appropriate use of programming structures (loops, conditionals, classes etc.) Poorly structured and inefficient algorithms Rarely uses good object-oriented design
Functionality	40 points (36-40 points) Program fulfills all functionality All requirements were fulfilled	35 points (29-35 points) Program fulfills most functionality Most requirements were fulfilled	28 points (0-28 points) Program does not fulfill functionality Few requirements were fulfilled

	Extra effort was apparent		
Test	20 points (18-20 points) Comprehensive test plan	17 points (15-17 points) Good test plan included	14 points (0-14 points) No test plan included
Documentation	20 points (18-20 points) Excellent comments	17 points (15-17 points) Good comments	14 points (0-14 points) No comments

Submit Cancel

	Excellent approach discussion and references	Some approach discussion	No approach discussion
Overall Score	**Exceed** **90 or more**	**Meets** **70 or more**	**Does not meet** **0 or more**

Submit Files

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File | Record Audio

Comments

Paragrap ▼