CIS 2430 A2 README/Checklist

Name:	O'Neal Yako
UoG Email Address:	oyako@uoguelph.ca
Partner's name (if applicable)	No partner
Percentage of your submission that is taken from the starter code provided (approximately)	(~90%) taken and then edited/changed up.
How to run your application: provide the exact string to type to compile it, and run it. Must be runnable on the command line.	(MacOs/linux) javac -cp ".:mysql-connector-java-8.0.13.jar" *.java java -cp ".:mysql-connector-java-8.0.13.jar" Planner (windows) javac -cp ".;mysql-connector-java-8.0.13.jar" *.java java -cp ".;mysql-connector-java-8.0.13.jar" Planner

Notes for TAs (anything special we should know when grading your assignment)	Note you can create a new student, or search by ID#. Once you have searched for a student/created a new student please expand the window so you can see the full frame. Once you have chosen your student either by creating a new one or by searching for one, you will be taken to the Welcome Screen. The transcript editing functions are on the left, as well as the buttons for you to see your transcript and your planned courses. You can Add to your planned courses on the right. When adding to the Planned Courses, please make sure you fill out all fields and put a Decimal value for the Weighted field (i.e 0.1, 0.2, 0.3) You can drop a course from both the transcript/Planned courses in the middle. In the top center you will see two buttons. Show Requirements and Required Courses Left. Show Requirements will always should you the full list of required courses for your major, regardless of what you have already taken. Show Requirements Left show you all the required courses that are not on your transcript. If you add a required course to your transcript (make sure you have the exact same course code i.e CIS*1250) it will be removed from this list. At the bottom center you will see a button called 'Required Courses Left with Planned Accounted'. That button will show you a list of all the required courses for your major, excluding the ones on your transcript, and the ones on your list of Planned Courses. You will also see information relating to your GPA and credits accummulated/needed. The display for these stats will not be redrawn every time you add edit/add/ remove a course, howerver if you rerun the app you will see them changed.

Learning Outcomes	**3 examples from your code. File name, line number**
refactor and restructure class design for improved encapsulation, modularity, cohesion and coupling	1. MainGUI.java line 52 2. Degree.java line 79 3. Course.java line 113
demonstrate use of inheritance through super/sub classes as well as through the use of interfaces	1. BCG.java line 26 2. Degree.java line 170 3. AdminWindow.java line 9
demonstrate clear understanding event driven programming through well designed listeners and gui components	1. StudentWindow.java line 627 2. AdminWindow.java line 69 3. Planner.java line 22
demonstrate service-based error handling through a rich set of exception classes that communicate specific errors to client classes	1. Degree.java line 132 2. PrepStudentScript.java line 35 3. CourseCatalog.java line 83
create a repeatable testing suite and justify the choice of test cases	Used student creation and student search as testing suite. Used TestDB.java and Test.java and the provided class ConnectionDemo.java for development functionality testing.
design and create a graphical user interface that is learnable and usable	1. StudentWindow.java line 610 2. AdminWindow.java line 59 3. StudentWindow.java line 748
use inner classes, anonymous classes, and/or lambdas effectively	1. MyConnection.java line 191 2. Course.java line 35 3. BCG.java line 87

Required elements	**Examples from your code (File name, line number) - more than one example preferred**
Exceptions and try/catch loops	Degree.java line 132
Error prevention/handling (might also be try/ catch or might be input checking)	CourseCatalog.java line 83
Two different layout managers	MainGUI.java line 52, MainGUI.java line 64
Separate window/panel for administration	MainGUI.java line 52, MainGUI.java line 64

Listeners	StudentWindow.java line 627
Course class refactored and immutable	Course.java line 9
Attempt class created	Attempt.java
Classes in package	Course.java, CourseCatalog.java
Refactor Plan of Study (include how/where you provided the functionality if you eliminated POS)	
Database usage	StudentWindow.java line 85
Javadoc comments (the most complete examples)	StudentWindow.java line 388
Evidence of testing	Look @ SampleStudents in results.sql

INSTRUCTIONS FOR PARTNERS

Partners must each complete this sheet and both partners must submit. If you do not submit you will not get a grade for A2. Complete this sheet in such a way as to illustrated YOUR contribution to this assignment.