

CIS 2430 A2 README/Checklist

Name:	Ralph Arvin De Castro
UoG Email Address:	rdecastr@uoguelph.ca
Partner's name (if applicable)	
Percentage of your submission that is taken from the starter code provided (approximately)	0 %
How to run your application: provide the exact string to type to compile it, and run it. Must be runnable on the command line.	<p>Compile: javac *.java</p> <p>(JAR FILE was extracted)</p> <p>Run Student: java StudentViewer</p> <p>Run Administrator: java AdministratorViewer</p> <p>Run Both at the same time: java Planner</p> <p>Class files are in there (Judi said it's fine in the forums)</p>
Notes for TAs (anything special we should know when grading your assignment)	<p>Good luck on exams! And Thank you!</p> <ul style="list-style-type: none"> - My implementation file is: THSTandGEOGImplementation.pdf - Additional functionalities <ul style="list-style-type: none"> o Implemented VMC (View, Model, Controller) <ul style="list-style-type: none"> ▪ StudentViewer.java ▪ StudentController.java ▪ StudentModel.java ▪ AdministratorViewer.java ▪ AdministratorController.java ▪ AdministratorModel.java o New Classes <ul style="list-style-type: none"> ▪ (Transcript.java, whole file) <ul style="list-style-type: none"> • Transcript class was created to handle list of attempts (add, remove) ▪ (Planned.java, whole file) <ul style="list-style-type: none"> • Planned class was used to handle courses that are planned (Course + semesterPlanned) ▪ (PlanOfStudy.java, whole file) <ul style="list-style-type: none"> • PlanOfStudy was restructured to handle Planned courses (add, remove)

	<ul style="list-style-type: none"> ○ Provides a list all courses in the centre panel of border layout of GUI ○ Saves and closes the a plan user without terminating and allowing user to load or create student again ○ Thorough testing <ul style="list-style-type: none"> ▪ If student not in database (NoStudentException.java) ▪ If fields are not filled in (EmptyFieldException.java) ▪ Checks for null ▪ Checks for divisible by zero ▪ Checks for invalid number format ▪ Ignores strings in grades when calculating GPA

Learning Outcomes	3 examples from your code. File name, line number
refactor and restructure class design for improved encapsulation, modularity, cohesion and coupling	<p>(Course.java , whole file)</p> <ul style="list-style-type: none"> - Course class fields (Grade and semester was removed) <p>(Attempt.java, whole file)</p> <ul style="list-style-type: none"> - Attempt java was used to handle Course, grade and semester fields <p>(Transcript.java, whole file)</p> <ul style="list-style-type: none"> - Transcript class was created to handle list of attempts (add, remove) <p>(Planned.java, whole file)</p> <ul style="list-style-type: none"> - Planned class was used to handle courses that are planned (Course + semesterPlanned) <p>(PlanOfStudy.java, whole file)</p> <ul style="list-style-type: none"> - PlanOfStudy was restructured to handle Planned courses (add, remove)
demonstrate use of inheritance through super/sub classes as well as through the use of interfaces	<p>(Degree.java, whole file)</p> <ul style="list-style-type: none"> - Abstract superclass - Setting and getting degree title <p>Subclasses of degree</p> <p>(BCG.java, whole file)</p> <p>(CS.java, whole file)</p> <p>(SEng.java, whole file)</p> <ul style="list-style-type: none"> - Each has different requirements and number of required credits - Used super(); - Boolean meetRequirements differ

demonstrate clear understanding event driven programming through well designed listeners and gui components	(StudentViewer.java, starting at line 44) <ul style="list-style-type: none"> - Each button has different action listener (AdministratorViewer.java, 37, 60 ,65) - Each button has different action listener (StudentController.java, whole file) (AdministratorController.java, whole file)
demonstrate service-based error handling through a rich set of exception classes that communicate specific errors to client classes	(StudentViewer.java, 382) (StudentViewer.java, 290) (StudentViewer.java, 369) Exception classes (NoStudentException.java, whole file) (EmptyFieldException.java, whole file)
create a repeatable testing suite and justify the choice of test cases	(PlannedTest.java,33 ,45) <ul style="list-style-type: none"> - Test if constructor works properly (PlannedTest.java,58, 84) - Test if fields sets properly (PlannedTest.java,71, 97) - Test for Null courses and semester
design and create a graphical user interface that is learnable and usable	(StudentViewer.java, 29) <ul style="list-style-type: none"> - JFrame for main menu (StudentViewer.java, 188) - JFrame for welcome menu (StudentViewer.java, 233, 315) - JFrames for create and load student (AdministratorViewer.java) - JFrame for administrator
use inner classes, anonymous classes, and/or lambdas effectively	Inner Class (StudentViewer.java, 271) (StudentViewer.java, 352) Lambdas (StudentViewer.java, 44) (StudentViewer.java, 70) I have more lambdas used for adding action listener in my StudentViewer.java

Required elements	Examples from your code (File name, line number) – more than one example preferred
Exceptions and try/catch loops	Yes (StudentController.java, 22) (StudentViewer.java, 382) (StudentViewer.java, 290) (StudentViewer.java, 369)

	<p>Exception classes (NoStudentException.java, whole file) (EmptyFieldException.java, whole file)</p>
Error prevention/handling (might also be try/catch or might be input checking)	<p>Yes (StudentViewer.java, 279, 350) Checking if user input an integer number for student number (StudentViewer.java, 280) Checking if user input at least a character for first name and last name (StudentViewer.java, 362) Checking if a student is in a database when a user tries to log in + many moreeee</p>
Two different layout managers	<p>Yes BorderLayout (StudentViewer.java ,32) GridLayout (StudentViewer.java, 60) Own Layout (StudentViewer.java, 233, 321)</p>
Separate window/panel for administration	<p>Yes Run - java AdministratorViewer</p>
Listeners	<p>Yes ActionListener Classes (StudentViewer.java, 271) (StudentViewer.java, 352) Lambdas (StudentViewer.java, 44) (StudentViewer.java, 70) I have around 20 lambdas used for adding action listener in my StudentViewer.java</p>
Course class refactored and immutable	<p>Yes (Course.java, whole file)</p>
Attempt class created	<p>Yes (Attempt.java ,whole file)</p>
Classes in package	<p>Yes (Course.java, whole file) (CourseCatalog.java, whole file) -package univ;</p>
Refactor Plan of Study (include how/where you provided the functionality if you eliminated POS)	<p>Yes (PlanOfStudy, wholefile) -used to handle planned courses</p>
Database usage	<p>Yes</p>

	(DBConnect.java. wholefile)
Javadoc comments (the most complete examples)	Yes (Course.java, wholefile) (Attempt.java, whole file)
Evidence of testing	Yes (PlannedTest.java,33 ,45) - Test if constructor works properly (PlannedTest.java,58, 84) - Test if fields sets properly (PlannedTest.java,71, 97) - Test for Null courses and semester

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.