CIS 2430 A2 README/Checklist

|  |  |
| --- | --- |
| Name: | Thomas Arsenault |
| UoG Email Address: | tarsen01@uoguelph.ca |
| Partner’s name (if applicable) | Reza Parnian |
| 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. | On Windows systems, you can run the following command from a command prompt window opened in the root directory of the project:  java -cp .;database/mysql-connector-java-8.0.13.jar Planner  On Linux or UNIX systems, you can use the provided makefile in the root directory to build the project with the “make build” command and to run it with the “make run” command with a terminal window opened in the root directory of the project. |
| Notes for TAs (anything special we should know when grading your assignment) | This whole assignment was done using mob programming, where both of us were actively programming on the same screen together, hence our Read Me files are identical as well.  You can find a help menu on the menu bar in StudentMenu.fxml and AdminMenu.fxml where you can open a README that instructs you how to complete each user story within the GUI.  Course codes must be entered with the following format: [CODE]\*[NUMBER], such as CIS\*2500  javafx.fxml.FXMLLoader$ValueElement processValue  WARNING: Loading FXML document with JavaFX API of version 10.0.1 by JavaFX runtime of version 8.0.191 may appear when running Planner depending on the version of JavaFX on your computer. It doesn’t affect anything! |

|  |  |
| --- | --- |
| **Learning Outcomes** | **3 examples from your code. File name, line number** |
| refactor and restructure class design for improved encapsulation, modularity, cohesion and coupling | Every degree class (BCG, BCH, etc) uses encapsulation to control access from subclasses and other classes  Due to the separation of major and degree title, there was a major restructuring of our degree classes  Course.java, grade and semester fields and their associated methods were moved to Attempt.java  PlanOfStudy.java functionality was moved to Student.java  CourseCatalog.java’s parser and loading methods were changed to read from an ArrayList of strings |
| demonstrate use of inheritance through super/sub classes as well as through the use of interfaces | Degree, major and the degree classes such as BCG, BCH demonstrate inheritance  All classes implement serializable interface |
| demonstrate clear understanding event driven programming through well designed listeners and gui components | StudentController.java:290 handleMenuClicks()  MainMenuController.java:30 handleButtonClicks()  AdminController:113 handleNewCourseClicks |
| demonstrate service-based error handling through a rich set of exception classes that communicate specific errors to client classes | LaunchDialog.java to launch an error message in an alert box in the GUI  Did not find it wise to use exceptions the user will never see them (since we used JavaFX, where the most effect way to show errors is the dialog box) |
| create a repeatable testing suite and justify the choice of test cases | In addition to the partial test cases on the moodle website, we tested our code with series of print statements using a wide range of input values (corner cases, edge cases, etc) |
| design and create a graphical user interface that is learnable and usable | NewMenu.fxml  AdminMenu.fxml  StudentMenu.fxml |
| use inner classes, anonymous classes, and/or lambdas effectively | Inner Classes: StudentController.java:270  Anonymous Classes:  Lambdas: StudentController.java:693  Student.java: Lots of lambdas everywhere |

|  |  |
| --- | --- |
| **Required elements** | **Examples from your code (File name, line number) – more than one example preferred** |
| Exceptions and try/catch loops | StudentController.java:666-691 |
| Error prevention/handling (might also be try/catch or might be input checking) | CourseCatalog.java:254 |
| Two different layout managers | StudentMenu.fxml:28, VBox  StudentMenu.fxml:485, GridPane |
| Separate window/panel for administration | AdminMenu.fxml  MainMenuController.java:37 |
| Listeners | StudentController.java:290  handleMenuClicks()  MainMenuController.java:30 handleButtonClicks()  AdminController.java:113  handleNewCourseClicks() |
| Course class refactored and immutable | Course.java |
| Attempt class created | Attempt.java |
| Classes in package | univ |
| Refactor Plan of Study (include how/where you provided the functionality if you eliminated POS) | Eliminated PlanOfStudy.java  Implemented as a ArrayList in Student.java |
| Database usage | StudentController.java:281 loadCatalog() |
| Javadoc comments (the most complete examples) | Degree.java |
| Evidence of testing | Administrator.java:80 main() |

**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.