CIS 2430 A2 README/Checklist

|  |  |
| --- | --- |
| Name: | Patrick Copp |
| UoG Email Address: | [pcopp@uoguelph.ca](mailto:pcopp@uoguelph.ca) |
| Partner’s name (if applicable) | No Partner |
| 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. | Directory should be pcopp:  Database is not used so do not compile with the sql.jar file  javac Planner.java  java Planner |
| Notes for TAs (anything special we should know when grading your assignment) | You must add to planner before adding to transcript. The view menu tab contains many of the user stories. Press update button to find current GPA, CIS GPA, Last 10 GPA, and whether you can graduate. Last 10 GPA shows GPA if there are less than 10 in transcript. Prerequisites for all in plan and for all in degree are not included as you can view all prerequisites for any course from the view menu. Database was not used at all for this program, the code will instead use text files to read the course catalog and save/load students. DO NOT EDIT THE TEXT FILES or it will break. I have basically used text files within pcopp as the database. Use the menubar File to save/load or load from the start of the program when it asks you. |

|  |  |
| --- | --- |
| **Learning Outcomes** | **3 examples from your code. File name, line number** |
| refactor and restructure class design for improved encapsulation, modularity, cohesion and coupling | Student.java: 13 created AttemptList, PlanList to replace planofstudy.  Attempt.java (all lines): created Attempt.java, this differentiates between an attempt made by a student and an available course to take  Course.java: 43, all other protected methods.  This protects all available courses from being modified by a user. |
| demonstrate use of inheritance through super/sub classes as well as through the use of interfaces | BCG.java  CS.java  SEng.java |
| demonstrate clear understanding event driven programming through well designed listeners and gui components | File: Planner.java  This entire file contains the GUI with numerous listeners and GUI components. |
| demonstrate service-based error handling through a rich set of exception classes that communicate specific errors to client classes | All files in Exceptions package:  BadFile.java  BadGrade.java  NullDegree.java |
| create a repeatable testing suite and justify the choice of test cases | I utilised the VPL provided on moodle to complete unit testing in my program. Regardless, there is no way to damage the program using the GUI. |
| design and create a graphical user interface that is learnable and usable | File: Planner.java  This entire file contains the GUI with numerous listeners and GUI components. |
| use inner classes, anonymous classes, and/or lambdas effectively | Planner.java line 414  Planner.java line 335  Planner.java line 287 |

|  |  |
| --- | --- |
| **Required elements** | **Examples from your code (File name, line number) – more than one example preferred** |
| Exceptions and try/catch loops | Planner.java line 81  Planner.java line 118  Student.java line 261 |
| Error prevention/handling (might also be try/catch or might be input checking) | Student.java line 41  Student.java line 64  Student.java line 79 |
| Two different layout managers | Planner.java line 132  Planner.java line 100 |
| Separate window/panel for administration | The administrator has access to the courselist catalog through the included text file courseListA2.txt, using that file courses can be modified and added/removed. The administrator can also change what courses are available by using the administrator button to insert a new course catalog file to use. The program will default to my included “courselistA2.txt” though upon reopening. I have included a course catalog called test for you to see the functionality. |
| Listeners | Planner.java line 428  Planner.java line 413  Planner.java line 376 |
| Course class refactored and immutable | Course.java line 51  Course.java line 93  Read class variables of Course to see refactoring. |
| Attempt class created | Attempt.java |
| Classes in package | I am assuming you are referring to univ, in that case Attempt, Course, CourseCatalog and FakePlanner.java are in univ. |
| Refactor Plan of Study (include how/where you provided the functionality if you eliminated POS) | I removed POS entirely and created two Attempt ArrayLists in Student, one for transcript (AttemptList) and one for Plan (PlanList). |
| Database usage | I did not use the database but you can still save/load and access the course catalog using text files (course catalog text file is included). |
| Javadoc comments (the most complete examples) | Course.java has complete Javadocs. |
| Evidence of testing | TESTING: getFullName  TESTING: getFirstName  testtoString(StudentTest): null  setLastName\_Null(StudentTest): Expected exception: java.lang.Exception  setFirstName\_Null(StudentTest): Expected exception: java.lang.Exception  false  TESTING: getFullName  TESTING: getFirstName  testtoString(StudentTest): null  setLastName\_Null(StudentTest): Expected exception: java.lang.Exception  setFirstName\_Null(StudentTest): Expected exception: java.lang.Exception  false  Moodle VPL. |

**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.