ITEC 3150, Homework #1 - Intermediate Programming Review

Coding standards note: The coding standards are in a document in a D2L module titled Coding Standards. See Grading Criteria for points lost if not followed.

Create a listing of available courses for a small school. The listing should contain a list of all courses. The user should be able to view a list of courses in the listing, search for a particular course by CRN or course name, delete a course found by the CRN or name, and add a course by providing all necessary information for that course. The addition and deletion should be visible the next time the program is run.

The courses should fall into three disciplines: English, Math, and History.

* All courses should have a unique CRN (numeric code) , name, category (English, Math, History).
* English courses additionally have two attributes:
  + level (freshman, sophomore, junior, senior)
  + They can be either a reading or writing course or both.
* Math courses have two additional attributes
  + STEM or non-STEM (but not both)
  + They can be online, in person or hybrid courses.
* History courses have two additional attributes
  + Area E eligible or not
  + They can be recorded, online, or in-person.

The courses are always read from a text file each time the program is run. The format of the text file is your choice and you MUST turn in a text file with a minimum of six courses. In order to receive credit for deleting or adding a courses, the program must also rewrite the file upon exiting.

Interface to program may be textual or graphic, I expect to see a menu of some sort with the ability to view courses, find a course, add a course, and delete a course and exit the program.

User interface must be minimally user-friendly.

To receive full credit for the program, it must use inheritance for the different types of courses.

Turn in: All Java files – should be a minimum of 4(four) separate class files. More is fine. Text file containing your minimum 6 courses.

Competencies being graded:

1. Ability to read from a text file
2. Ability to use inheritance to create a set of subclasses all inheriting from a common parent class.
3. Using polymorphism to create and maintain a single list of the subclasses
4. Use System.out.println and Scanner to create a user friendly interface into your program
5. Ability to read an English language problem description and design a multi class solution in Java
6. Ability to follow given coding standards- in D2L content under Coding Standards.

Grading Criteria

|  |  |
| --- | --- |
| Compiles and has minimum of 4 separate Java classes and files: | 10 |
| Correctly uses inheritance for courses | 10 |
| Correctly lists courses available | 15 |
| Correctly finds (or doesn’t find) courses | 10 |
| Correctly deletes courses | 10 |
| Correctly add courses | 10 |
| Correctly reads text file | 10 |
| Correctly writes text file (upon deletion or insertion) | 10 |
| Meets coding standards | 15 |