Implemented a generic Feedback class with a public static array list of all given feedback and a loop that parses through it and verifies the professor of the course corresponding to each individual feedback with the professor trying to view it before giving them clearance. Only allows students to give feedback for courses they have registered for (again by verifying course code provided by student via a loop). Student can either rate the course from 1-5 (rating is saved as an integer) or press 0 to give descriptive feedback (stored as a string).

Implemented a TA class that inherits all the properties of the student class and added two functionalities (viewing and updating their grades) to it. In the main program, compared the TA's email ID to the assigned TA of the course and if they matched (verifying that the user is in fact the TA of the course), typecasted the user object's class from Student to TA (downcasting). TA needs to be set by the professor of the course (who needs to be set by the admin).

Added 3 exception classes for the 3 required exceptions and implemented them in their respective functions as well as in a try-catch block in the main function. Initialized course capacity to 1 for convenience of testing so that it automatically becomes impossible to register after one student has registered, meaning you only need to go through two iterations of the program to see the CourseFullException in action. For convenience of testing of the DropDeadlinePassedException exception, it decrements the days to the deadline parameter (initialized to 3) with each iteration of the portal considered a new day so the person running the code can see how it works at first but not after a few iterations (which represent the passing of a few days). For the real portal, it would decrement at midnight every day instead of at every iteration.