Project 2

Nelson Villatoro

CMSC215

February 5, 2025

**Program Goals & Objectives**

The primary goal of this program is to read student data from a file and then calculate each student’s GPA, which will be used to determine if the student is eligible for honor society membership based on a GPA cutoff threshold (equal to the average GPA of all valid students combined with the highest possible GPA (which is 4.0)). Individual students who satisfy the GPA threshold will be listed in the final output of the program.

Program Development Schedule:  
  
Day 1: Student and Undergraduate Class

Day 2: Graduate Class

Day 3: Project2 Class

Day 4: Project2 Class

Day 5: Final refinements and testing

Day 6: Finalization of program

**Note:** Unique rules apply for Undergraduate vs Graduate students. See further specification below.

**Design Overview**

Program structure (4) Classes:

1. **Student**

• **Purpose**: Base(super) class for all students, storing name, credit hours, and total quality points.

• **Key Methods**:

• gpa(): Calculates a student’s GPA by dividing total quality points by credit hours.

• eligibleForHonorSociety(): Compares the student’s GPA to a static gpaThreshold.

2. **Undergraduate**

• **Inheritance**: Extends Student.

• **Unique Feature**: Includes a Year enum (FRESHMAN, SOPHOMORE, JUNIOR, SENIOR), restricting honor society membership to JUNIOR or SENIOR if the GPA meets the threshold.

3. **Graduate**

• **Inheritance**: Extends Student.

• **Unique Feature**: Includes a degreeType field (e.g., “Masters” or “Doctorate”), limiting honor society eligibility to those with a “Masters” degree type who meet the threshold.

4. **Project2**

• **Driver Class**: Reads student data from an input file for processing and constructing the appropriate objects (Undergraduate or Graduate), calculates the honor society GPA threshold, and prints the results of students who qualify.

**Program Functional Requirements**

**Program Functional Requirements**

1. **File Input**

• The program reads data lines from students.txt.

• Each line must have at least four tokens: (1) student name, (2) credit hours, (3) quality points, and (4) classification token (e.g., freshman, senior, masters, etc.).

2. **Object Creation**

• If the classification token indicates an undergraduate year, an Undergraduate object is created.

• Otherwise, a Graduate object is created.

3. **Storing Data & GPA Calculation**

• All valid Student objects are added to an ArrayList.

• The program calculates each student’s GPA (if credit hours > 0) and tracks the average GPA across all valid students.

4. **Determining Threshold**

• The honor society threshold is the midpoint between the average GPA and 4.0:

threshold = (averageGPA + 4.0) / 2.0

5. **Eligibility Check**

• **Undergraduates**: Must meet/exceed the threshold and be either JUNIOR or SENIOR.

• **Graduates**: Must meet/exceed the threshold and have a “masters” degree type (case-insensitive).

6. **Output Results**

• Displays the calculated GPA threshold.

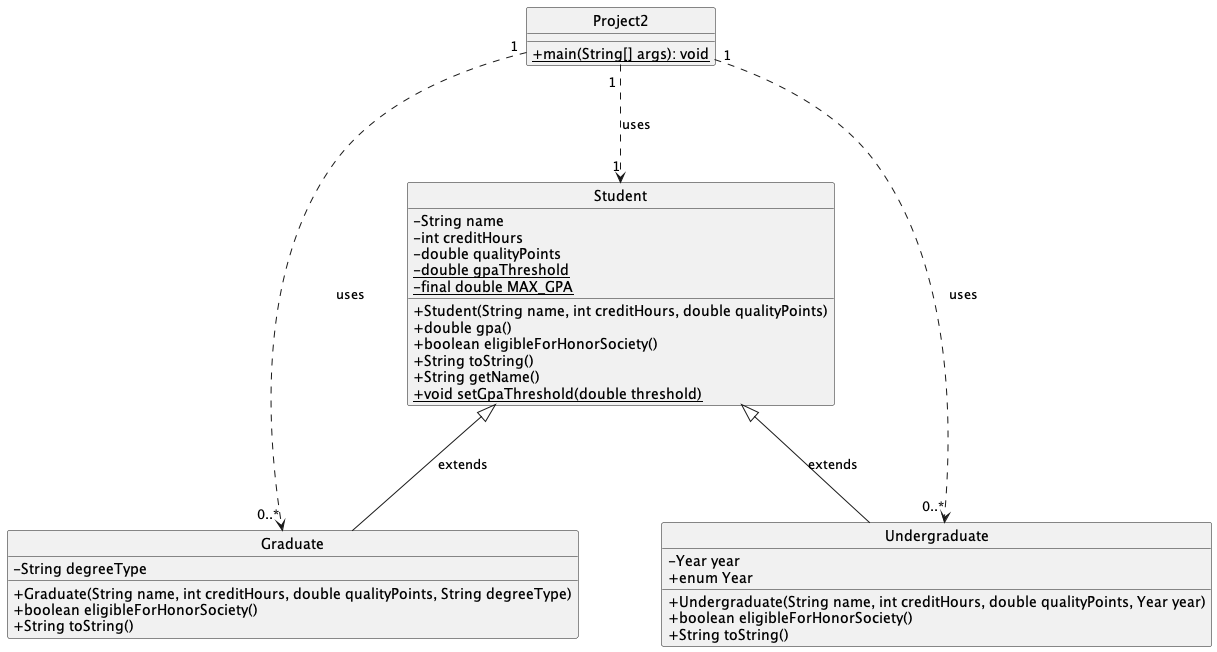
• Lists each qualifying student (showing name, GPA, classification/degree).

7. **Termination**

• If students.txt is missing, the program shows “File Not Found” and terminates.

• Otherwise, it processes to the end of the file, outputs the results, and then terminates as expected.

**Honor Society Eligibility UML Class Diagram**

****

**Testing**

I performed testing to verify the program's functionality. Below is a traceability matrix that details the unit tests conducted to ensure proper parsing of student data, proper GPA calculation, proper identification of honor society-qualified students, and verify that the program handles edge cases correctly and terminates as expected in a user friendly manner.

**Table 1 – Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input/Output | Expected Result | Actual Result | Outcome  (Pass/Fail) |
| TC-01 | Troy 45 154 Freshman | GPA threshold set to individual student’s GPA, and nobody is eligible | GPA threshold for membership is 3.71  (Empty list of students eligible) | Pass |
| TC-02 | Torres 45 171 Junior  Smith 15 45 Freshman  Johnson 30 90 Sophomore | GPA threshold for membership is 3.63  Name: Torres GPA 3.80 JUNIOR | GPA threshold for membership is 3.63  Name: Torres GPA 3.80 JUNIOR | Pass |
| TC-03 | Smith 0 0 Junior  Johnson 60 162 Graduate  Davis 45 176 Masters  Ashley 54 155 Junior | Should handle division by 0 edge case and continue functioning | GPA threshold for membership is 3.58  Name: Davis GPA 3.91 MASTERS | Pass |
| TC-04 | No file to read | User should be prompted proper troubleshooting to handle file not found | “File not found.” | Fail |

**Test Output Screenshots**

**TC-01**

**A screen shot of a computer

AI-generated content may be incorrect.**

**TC-02**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**TC-03**

A screen shot of a computer

AI-generated content may be incorrect.

**TC-04**

A black screen with white text

AI-generated content may be incorrect.

**Brief Note on Lessons Learned:**

This project taught me the importance of handling edge cases—such as zero credit hours or a missing text file- highlighting the need to develop software that can deal with real-world data inputs. Variation in how users interact with software applications introduces the possibility of program crashes; as such, learning how to implement exception handling has been a valuable lesson to learn moving forward with my software development career.