Sened Abraham Desalegn

Studid: 20082

Home work 4

**Introduction**

The GradeBookApp is a command-line interface program designed to assist instructors in managing student grades efficiently. It allows instructors to set up grading policies, add student records, and input and manage grades for assignments, tests, and final exams. The application supports sorting student information by name or ID and provides functionalities to calculate final grades based on predefined grading weights. All data, including student details and grading policies, are saved and managed through system files, making it a comprehensive tool for academic grading management.

# **Test Cases and Results**

**Test Case Structure**:

* Each test case should include:
  + - Description: A brief overview of what the test case is testing.
    - Expected Outcome: What should happen when the test case is executed.
    - Actual Outcome: The real result of executing the test case.
    - Screenshots showing the test case execution.

**Documenting Test Cases**:

Below I have created several test cases for the major functionality of the application. You may reproduce each test case as specified below.

**Gradebook Instructor System**

**1.Test Case ID: TC001**

* **Description:** Set up a new semester in the grade book system.
* **Preconditions:** The application is running.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to set up a new semester ('S').

Enter the number of programming assignments, tests, and final exams.

Enter the weight percentages for assignments, tests, and final exams.

Confirm the setup.

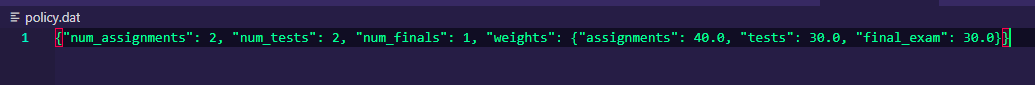
* **Expected Outcome:** The new semester setup is confirmed and saved in the system.
* **Actual Outcome:** The new semester setup is confirmed and saved in the system.
* **Status:** Success
* **Screenshots:**

A screenshot of a computer program

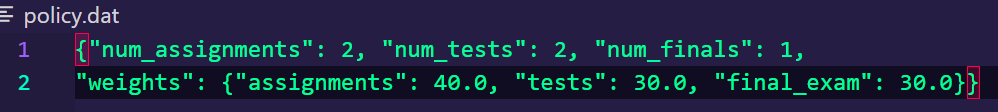
Description automatically generatedA computer screen with white text

Description automatically generated

The information is saved in the polic.dat file as shown:



For visibility:



* **Test Case ID:** TC002
* **Description:** Add a new student to the grade book.
* **Preconditions:** The semester has been set up.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to add a student ('A').

Input the student’s ID, first name, and last name.

Confirm the addition of the student.

* **Expected Outcome:** The new student is added and displayed in the list of students.
* **Actual Outcome:** The new student is added and displayed in the list of students.
* **Status:** Success

A computer screen with white text

Description automatically generated

A blue and white text on a dark background

Description automatically generated

A computer code with blue text

Description automatically generated

* **Test Case ID:** TC003
* **Description:** Record grades for a programming assignment.
* **Preconditions:** At least one student has been added to the system.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to record programming assignment grades ('P').

Enter the assignment number for which grades are to be recorded. (We had two assignments two option (1,2):

Input grades for each student.

Confirm the recording of grades.

* **Highlighted point:** The maximum score for an assignment is calculated to ensure that the score entered for each student does not exceed the proportionate value that each assignment contributes towards the final grade, maintaining consistency across all assignments.
* **Expected Outcome:** Grades are recorded and updated in the students' records.
* **Actual Outcome:** Grades are recorded and updated in the students' records.
* **Status:** Success
* **Screenshots:** (I have only added one student earlier in Test Case TC002, for elaboration I will add another student following the same step as TC002):

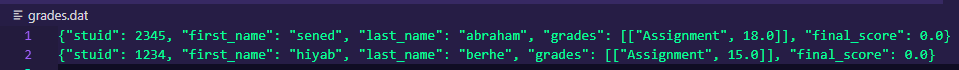
A computer screen with white text

Description automatically generated

A screenshot of a computer code

Description automatically generated

The info is also saved in “Grades.dat”



More Clarified:

A screenshot of a video game

Description automatically generated

* **Test Case ID:** TC004
* **Description:** Record grades for a test.
* **Preconditions:** At least one student has been added to the system.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to record test grades ('T').

Enter the test number for which grades are to be recorded. (We had two assignments two option (1,2)

* Highlighted point: The maximum score for an assignment is calculated to ensure that the score entered for each student does not exceed the proportionate value that each assignment contributes towards the final grade, maintaining consistency across all assignments.

Input grades for each student.

Confirm the recording of grades.

* Expected Outcome: Test grades are recorded and updated in the students' records.
* Actual Outcome: Test grades are recorded and updated in the students' records.
* Status: Success

A computer screen with white text

Description automatically generated

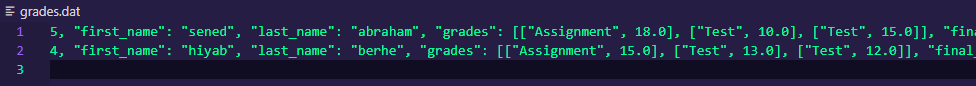
A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

Then they are also updated in the “grades.dat”



More clarified pic:

A computer screen shot of a test

Description automatically generated

* **Test Case ID:** TC005
* **Description:** Record grades for a Final Exam.
* **Preconditions:** At least one student has been added to the system.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to record test grades ('F').

Enter the test number for which grades are to be recorded. (We had two assignments 1 option (1)

* **Highlighted point:** The maximum score for an assignment is calculated to ensure that the score entered for each student does not exceed the proportionate value that each assignment contributes towards the final grade, maintaining consistency across all assignments.

Input grades for each student.

Confirm the recording of grades.

* **Expected Outcome:** Test grades are recorded and updated in the students' records.
* **Actual Outcome:** Test grades are recorded and updated in the students' records.
* **Status:** Success

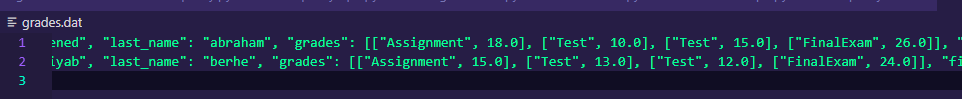
A computer screen with white text

Description automatically generated

A screenshot of a computer code

Description automatically generated

Also saved to “grades.dat”



More Clarified:

A screenshot of a computer program

Description automatically generated

* **Test Case ID:** TC006
* **Description:** Change a student's grade.
* **Preconditions:** Grades have been recorded for assignments or tests.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to change a grade ('C').

Input the student’s ID and the type of grade to change.

Enter the new grade and the specific assignment/test number.

Confirm the grade change.

* **Expected Outcome:** The grade change is reflected in the student's record.
* **Actual Outcome:** The grade change is reflected in the student's record.
* **Status:** Success

--Testing to change Assignments

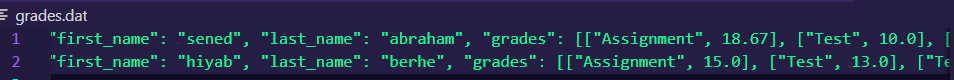
A computer screen with white text

Description automatically generated

A blue and white text on a black background

Description automatically generated

Successfully, updated in the file.



A computer screen shot of a computer code

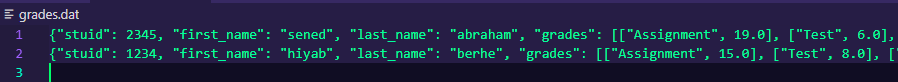
Description automatically generated

Testing to update Tests:

A blue and green text on a black background

Description automatically generated

Updates successfully in the file “grades.dat”



A screenshot of a computer program

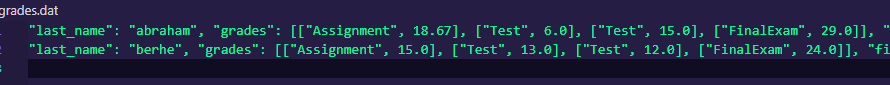
Description automatically generated

Testing to update Finals:

A blue and white text on a black background

Description automatically generated

Updates successfully in the file “grades.dat”



A screenshot of a computer program

Description automatically generated

* **Test Case ID:** TC007
* **Description:** Calculate final scores for all students.
* **Preconditions:** All necessary grades (assignments, tests, final exam) have been entered.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to calculate final scores ('G').

Confirm the calculation.

* **Expected Outcome:** Final scores are calculated based on defined weights and updated in the students' records.
* **Actual Outcome:** Final scores are calculated based on defined weights and updated in the students' records.

Status: Success

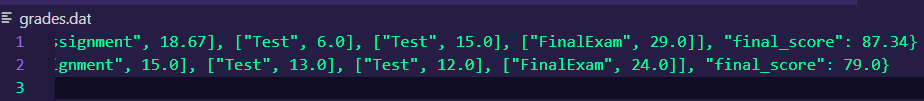
A computer screen with white text

Description automatically generated

A blue and white text

Description automatically generated

Updates successfully in the grades.dat file



A screenshot of a computer program

Description automatically generated

* **Test Case ID:** TC008
* **Description:** Output the grade data.
* **Preconditions:** Final scores have been calculated.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to output grade data ('O').

Displays output both on ID and By name based on the selected option.

* **Expected Outcome:** Grade data is output in the desired format and displayed.
* **Actual Outcome:** Grade data is outputted in the desired format and displayed.
* **Status:** Success

A computer screen with white text

Description automatically generated

It sorts the Names based on the last names.

A computer screen shot of a program

Description automatically generated

Sorting by Id:

A screenshot of a computer program

Description automatically generated

* **Test Case ID:** TC009
* **Description:** Quit the application.
* **Preconditions:** The application is running.
* **Test Steps:**

Navigate to the main menu of the GradeBookApp.

Select the option to quit ('Q').

* **Expected Outcome:** The application saves any changes and closes.
* **Actual Outcome:** The application saves any changes and closes.
* **Status:** Success

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Some Important Cases Handled:

1. When there is no policy, no commands can be executed. They should be set first.

A blue and green text

Description automatically generated

1. When the user enters the command “S”, all data in the files will be ignored. And the new policy will be included for one semester.
2. While running, is the instructor finished and want to set for another semester. They will reieve prompt as follows below:

A screenshot of a computer screen

Description automatically generated

1. There are many validations to check if the data inputs are correct. For example, while adding a student the ID should be between 1-9999. If it is out of range, it throws exception through the error handling.

A blue and green text

Description automatically generated