Testing Report

By Seah Ying Xiang 謝穎翔 (T12902136) t12902136@ntu.edu.tw

> National Taiwan University
>
> EE5191 Software Testing & Security Checking
>
> Project 1: White-box unit testing with CPP and coverage instrumentation

Testing Environment

Ubuntu 22.04.3 LTS on WSL 2.0

Prerequisite Packages

- g++
  - `sudo apt install g++`
- make
  - `sudo apt install make`
- libcppunit-dev
  - `sudo apt install libcppunit-dev`
- gcovr
  - `sudo apt install gcovr`

Testing Procedure

Ensure that all prerequisite packages have been installed.

1. If the project root folder contains the `test` file or `*.gcda` or `*.gcno` files, run the following command to clean up the test files:

   ```
   make clean
   ```

2. To generate the test files and coverage reports, run the following command:

   ```
   make
   ```

   or

   ```
   make test
   ```

   The test will automatically execute, followed by the coverage reports.

Explanation of Software Under Testing (SUT)

File	Description
Course.cpp	Records course ID and grade score, and retrieves course name using a map from ID to name.

File	Description
Student.cpp	Records the student name and id, as well as their assigned courses.

Specification

- `src/Course.cpp` - `Course`
  - **Course id** should be a string that matches one of the predefined course IDs in `COURSE_MAP` defined in `src/CourseList.h`.
  - **Course name** should be a string that correspond to the value mapped from the ID key in `COURSE_MAP`.
  - **Course grade score** should be an integer from -1 to 100. -1 is used to denote that the grade score is unassigned/not applicable for this course.
  - **Course grade letter** should be one of the following strings: `"INV"`, `"NA"`, `"A"`, `"B"`, `"C"`, `"D"`, `"E"`, `"F"`.

Grade Score	Grade Letter
90 ≤ x ≤ 100	A
80 ≤ x < 90	B
70 ≤ x < 80	C
60 ≤ x < 70	D
50 ≤ x < 60	E
0 ≤ x < 50	F
-1	NA
Others	INV

- `src/Student.cpp` - `Student`
  - **Student name** should be a string.
  - **Student ID** should be a string.
  - **Add course** - A course can be added only if the student has less than `MAX_COURSES` number of courses assigned.
  - **Drop course** - A course can be dropped only if the student already has the course assigned.
  - **Assign grade score** - Grade score of a course can be assigned only if the student already has the course assigned.
  - **Retrieve grade score** - Grade score of a course can be retrieved only if the student already has the course assigned.
  - **Retrieve grade letter** - Grade letter of a course can be retrieved only if the student already has the course assigned.
  - **Number of assigned courses** is an integer from 0 to `MAX_COURSES`.

- **Get course by index** - A course object can only be retrieved only if the index is between 0 and number of assigned courses (non-inclusive).

Test Requirements

The program should allow a `name` string and `id` string to be assigned to a `Student` object, which can add and drop `Course` objects, as well as retrieve and assign grade scores and letters to each `Course` object assigned to the `Student` object.

The `Course` object should be able to map its `course id` to the course name based on the `COURSE_MAP` map defined in `src/CourseList.h`.

Test Plan

All test cases are defined in the test fixture in `src/TestStudent.h` and `src/TestStudent.cpp`

Test Cases

1. `testConstructor` - Test `Student` object constructor
    1. Create `Student` object with `name` and `id`.
    2. Verify the `Student` object has the assigned `name` and `id`.
2. `testAssignNoInitialGradeScore` - Test assigning courses without an initial grade score, verifying the default grade score
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object without assigning an initial grade score.
    3. Verify that the assigned courses should have an initial grade score of `-1`.
    4. Verify that invalid/unassigned courses should return a grade score of `-100`.
3. `testAssignNoInitialGradeLetter` - Test assigning courses without an initial grade score, verifying the default grade letter
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object without assigning an initial grade score.
    3. Verify that assigned courses should have initial grade letter of `"NA"`.
    4. Verify that invalid/unassigned courses should return a grade letter of `""`.
4. `testAssignInitialGradeScore` - Test assigning courses with an initial grade score
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object while assigning an initial grade score.
    3. Verify adding additional course beyond the max course number `MAX_COURSES` should not proceed.
    4. Verify that assigned courses should have the correctly assigned grade scores.
    5. Verify that invalid/unassigned courses should return a grade score of `-100`.
5. `testAssignInitialGradeLetter` - Test assigning courses with an initial grade score, but checking the grade letter
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object while assigning an initial grade score.
    3. Verify that assigned courses should have the correctly assigned grade letters.
    4. Verify that invalid/unassigned courses should return a grade letter of `""`.
6. `testAssignGradeScore` - Test assigning grade scores to courses after their construction
    1. Create `Student` object with `name` and `id`.

    2. Add a few courses to the `Student` object without assigning an initial grade score.
    3. Assign a grade score to the assigned courses.
    4. Verify the assigned grade scores.
    5. Verify that trying to assign a grade score to an invalid/unassigned course will fail
7. `testAddAndDropCourse` - Test adding and dropping courses
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object while assigning initial grade scores.
    3. Verify that the courses should have the correctly assigned grade scores.
    4. Drop a course, then verify that trying to retrieve the grade score of the dropped course should now return `-100`, while the other courses still return their assigned grade scores.
    5. Repeat the previous step until all courses are dropped.
8. `testGetAssignedCourseDetails` - Test retrieving an assigned course object and getting its details
    1. Create `Student` object with `name` and `id`.
    2. Add a few courses to the `Student` object while assigning initial grade scores.
    3. Verify the number of courses assigned to the student.
    4. For each course assigned, do the following:
        1. Retrieve the assigned `Course` object by its index in the `Student` object's course list.
        2. Verify the course name by comparing it against the `COURSE_MAP`.
        3. Verify the grade score is as assigned.
9. `testSetCourseGradeScore` - Test assigning a grade score to a course object after its construction
    1. Create a `Course` object with a valid `course id` (any of the course ids defined in `COURSE_MAP` in `src/CourseList.h`) and initial grade score of `-1`.
    2. Verify the name, id, grade score, and grade letter of the course matches the initial assigned values from the constructor.
    3. Set the grade score of the `Course` object to another value from 0 to 100.
    4. Verify that the retrieved grade score matches the assigned value.
    5. Verify that the retrieved grade letter matches the score value.

Test Result

Before Bug Fix

```
!!!FAILURES!!!
Test Results:
Run:  9   Failures: 5   Errors: 0


1) test: StudentTest::testAssignNoInitialGradeScore (F) line: 56 src/TestStudent.cpp
equality assertion failed
- Expected: -100
- Actual  : -1


2) test: StudentTest::testAssignNoInitialGradeLetter (F) line: 83 src/TestStudent.cpp
equality assertion failed
- Expected:
- Actual  : INV


3) test: StudentTest::testAssignInitialGradeScore (F) line: 127 src/TestStudent.cpp
equality assertion failed
- Expected: -100
- Actual  : -1


4) test: StudentTest::testAssignInitialGradeLetter (F) line: 164 src/TestStudent.cpp
equality assertion failed
- Expected: A
- Actual  : B


5) test: StudentTest::testAddAndDropCourse (F) line: 226 src/TestStudent.cpp
equality assertion failed
- Expected: -100
- Actual  : -1
```

```
------------------------------------------------------------------------------
                           GCC Code Coverage Report
Directory: .
------------------------------------------------------------------------------
File                                       Lines    Exec  Cover   Missing
------------------------------------------------------------------------------
src/Course.cpp                              27      26    96%   25
src/Student.cpp                             49      48    98%   45
src/TestMain.cpp                             8       8   100%
src/TestStudent.cpp                        169     140    82%   57-58,60,84-85,87,12
8-130,132,165-166,169-171,173,227-228,232-235,239-242,245-246,248
src/TestStudent.h                           11      11   100%
------------------------------------------------------------------------------
TOTAL                                      264     233    88%
------------------------------------------------------------------------------
```

Test Failure #	Bug Details
1	Trying to retrieve a grade score from an invalid/unassigned course returned a value of `-1` instead of `-100`
2	Trying to retrieve a grade letter from an invalid/unassigned course returned a value of `"INV"` instead of `""`

Test Failure #	Bug Details
3	Trying to retrieve a grade score from an invalid/unassigned course returned a value of `-1` insteald of `-100`
4	A course with a grade score of `90` should return a grade letter of `"A"`, but it returned `"B"` instead. Reason was the wrong equality sign used in line 24 of `src/Course.cpp`, should be `>=` instead of `>`.
5	Trying to retrieve a grade score from a dropped course returned a value of `-1` instead of `-100`.
6	Trying to retrieve a grade letter from a course with a invalid grade score of `110` returned `"A"` instead of `"INV"`.

After Bug Fix

```
OK (9 tests)



gcovr
------------------------------------------------------------------------------
                           GCC Code Coverage Report
Directory: .
------------------------------------------------------------------------------
File                                       Lines    Exec  Cover   Missing
------------------------------------------------------------------------------
src/Course.cpp                                29      29   100%
src/Student.cpp                               49      49   100%
src/TestMain.cpp                               8       8   100%
src/TestStudent.cpp                          169     169   100%
src/TestStudent.h                             11      11   100%
------------------------------------------------------------------------------
TOTAL                                        266     266   100%
------------------------------------------------------------------------------
```

Changes

1. `src/Student.cpp`
    1. Fixed return value of grade scores for invalid/unassigned courses from `-1` to `-100`.
    2. Fixed return value of grade letter for invalid/unassigned courses from `""` to `"INV"`.
2. `src/Course.cpp`
    1. Fixed comparator for the `"A"` letter grade from `>` to `>=`.
    2. Added check for grade scores `>100` to return a grade letter for `"INV"`.