

You must submit a zipped, **and only zipped**, archive of your homework directory, hw6. The directory must contain, at a minimum, the files `hw6/src/grade.cc`, `hw6/inc/grade.h`, `hw6/src/rational.cc` and `hw6/inc/rational.h`. Name the archive submission file `hw6.zip`

I will use my own makefile to compile and link to your `hw6/src/grade.cc`, `hw6/inc/grade.h`, `hw6/src/rational.cc` and `hw6/inc/rational.h` files. You must submit, at least, these four files.

DO NOT use additional dependencies. A Python script is used to extract your code and place it in a test directory. The script is not aware of additional dependencies your code might require.

Introduction

In this assignment, you will update the concept of the Grade class again, this time eschewing inheritance in favor of a template. You must also provide a Rational class to test your Grade class and list the necessary methods below. In this version, Rational is passed as a template to Grade and will likely require additional overloaded operators to implement the methods in Grade.

Any code you previously submitted that you believe is tested and working you may reuse in this assignment, but should consider this assignment as a complete rewrite—that is you do not have to cite your source if your previous code is your source.

Read the provided header file documentation for instructions on how the methods should work.

Points

Correctness

1. `Grade::Grade`: 1.05 points
2. `Grade::ToLetter`: 1.75 points
3. `Grade::CalcAverage`: 1.5 points
4. `operator<<(Rational)`: 1.75 points
5. `Rational::operator==()`: 1.45 points

Total: 7.5 Points Possible

Base Submission

- compilation of each test (ten tests above): 0.1 points each
- `hw5/src/grade.cc` style: 0.5 points
- `hw5/inc/grade.h` style: 0.5 points
- `hw5/src/rational.cc` style: 0.5 points
- `hw5/inc/rational.h` style: 0.5 points

Total: 2.5 Points Possible

Notes

I have provided you basic test apps which you can use to ensure that your code is, at least partially, correct. As always, I would suggest a more rigorous testing scheme to ensure that you test overloaded operators you require, but that I do not request.

Late assignments will be accepted with a 2.5% penalty per day late until 11:59 PM April 18th (-15%). No assignment will be accepted after this time. The assignment will be removed from Blackboard at that time. Any submission will be graded and if it is greater than the student's current Homework average, will be kept. Those below the average will be exempted.

This assignment may be used to replace one of Programming Assignments 4 or 5 at your discretion. You must let me know which within 24 hours of grading. After that time requests will be ignored.