COP 3331 Summer 2017: Programming Assignment 4
Due: Tuesday, 27 June, 11:55 pm

Please include the following files in a zipped folder and submit the zipped file via the assignment link on Canvas. The zipped file should have the name "proj4-xxx.zip" where xxx is your NetID.

- Inheritance Exercise (50 pts)
  - Modified Class File for GradedActivity (10 pts) – GradedActivity.h
    - For convenience, include the GradedActivity.cpp code (provided) within this class file
  - Class File for Essay (20 pts) – Essay.h
    - You may define functions inline for this headerfile
  - Driver Program (20pts) – EssayDriver.cpp

- README file: A plain text including instructions on how to compile and run your code in the IDE you used. This file should include any special instruction/information that the TA should know to be able to run your code.

<u>Inheritance Exercise</u>

Create an Essay class that is derived from the GradedActivity presented as the code example this week. The Essay class should determine the grade that a student receives on an essay. The student's essay score can be up to 100, and is determined in the following manner:

Grammar: 30 points
Spelling: 20 points
Correct length: 20 points
Content: 30 points

Demonstrate the class in a simple program.

You will need:

1. A base class, GradedActivity (provided on Canvas under Code Examples). Make changes to the file so that:
   a. The variable score is protected
   b. The getLetterGrade definition exists in this file
   c. Member functions are made virtual for any situations where a function is redefined (see 2d and 2e)
2. A derived class, Essay, which you will create. The derived class should have the following members (these can be defined inline):

    a. Four private members (variables) to store the points obtained in each of the four categories
    b. Accessors for each member
    c. Mutators for each member
    d. A redefined getScore function that adds the points from the four categories
    e. A redefined getLetterGrade function that
        i. calls getScore to assign a value to score (the member in GradedActivity)
        ii. Uses the getLetterGrade definition and returns the appropriate grade

3. A driver program that tests your derived class. Your driver should
    a. Prompt the user for each of the points in the categories (use input validation)
    b. Pass the values to your classes and display the total score, and the appropriate letter grade

Sample output is shown below:

```
How many grammar points did the student earn? (0 - 30): 280
Invalid. Enter a number from 0 through 30: 28
How many spelling points did the student earn?  (0 - 20): -17
Invalid. Enter a number from 0 through 20: 17
How many points for length did the student earn?  (0 - 20): 16
How many points for content did the student earn?  (0 - 30): 21
Grammar points: 28
Spelling points: 17
Length points: 16
Content points: 21
Overall score: 82
Grade: B
```