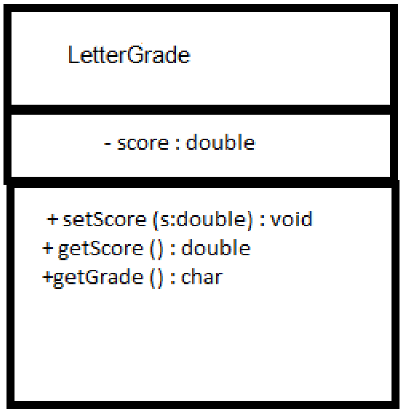
Lakehead University
Faculty of Science and Environmental Studies
Department of Computer Science
CS 2477 Object Oriented Programming
Practical Assignment No. 4

Q1: A graded activity can consist of a numeric score such as 70, 85, 90, and so on, and a letter grade such as A, B, C, D, or F. The following UML diagram for the LetterGrade class is designed to hold the numeric score and letter grade of a graded activity. When a numeric score is stored by the class, it automatically determines the letter grade. Method getGrade returns the letter grade that corresponds to the numeric score. So if (score >= 90) letterGrade = 'A'; else if (score >= 80) letterGrade = 'B'; and so on.



LetterGrade

- score : double

+ setScore (s:double) : void
+ getScore () : double
+getGrade () : char

In a course, a teacher gives the following tests and assignments:

• A lab activity that is observed by the teacher and assigned a numeric score.
• A pass/fail exam that has 10 questions. The minimum passing score is 70.
• An essay that is assigned a numeric score.
• A final exam that has 50 questions.

Design an Essay class that is **derived** from the of LetterGrade class presented above. The Essay class should determine the grade 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.

Write a class named CourseGrades. The class should have a member named grades that is an array of LetterGrade pointers. The grades array should have four elements, one for each of the assignments previously described. The class should have the following member functions:

setLab: This function should accept the address of a LetterGrade object as its argument. This object should already hold the student’s score for the lab activity. Element 0 of the grades array should reference this object.

setPassFailExam: This function should accept the address of a PassFailExam object as its argument. This object should already hold the student’s score for the pass/fail exam. Element 1 of the grades array should reference this object. A sample of the PassFailExam output is shown below:

```
How many questions are on the exam? 100 [Enter]
How many questions did the student miss? 25 [Enter]
Enter the minimum passing score for this test: 60 [Enter]
Each question counts 1.0 points.
The minimum passing score is 60.0
The student's exam score is 75.0
The student's grade is P
```

setEssay: This function should accept the address of an Essay object as its argument. Use a LetterGrade object. This object should already hold the student’s score for the essay. Element 2 of the grades array should reference this object.

setPassFailExam: This function should accept the address of a FinalExam object as its argument. This object should already hold the student’s score for the final exam. Element 3 of the grades array should reference this object. The FinalExam class should inherit methods setScore(double) , getScore(), and getLetterGrade() from LetterGrade. A sample of the FinalExam output is shown below:

```
How many questions are on the final exam? 20 [Enter]
How many questions did the student miss? 3 [Enter]
Each question counts 5 points.
The exam score is 85
The exam grade is B
```

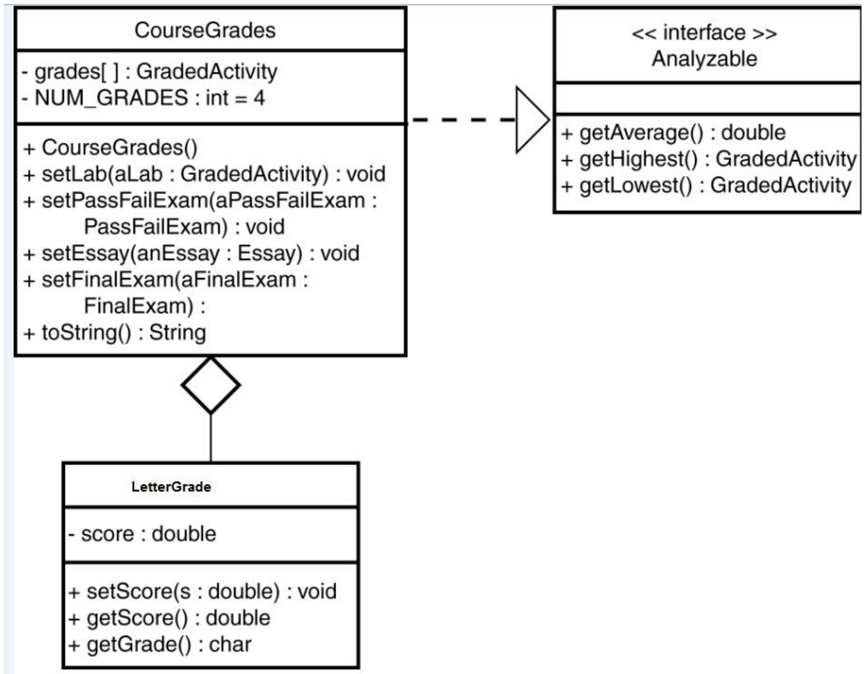
print: This function should display the numeric scores and grades for each element in the grades array.

Demonstrate the class in a program.

Q2: Modify the courseGrades class you created in the previous question so it implements the following **interface**:

```
public interface Analyzable
{
double getAverage();
GradedActivity getHighest();
GradedActivity getLowest();
}
```

The getAverage method should return the average of the numeric scores stored in the grades array. The getHighest method should return a reference to the element of the grades array that has the highest numeric score. The getLowest method should return a reference to the element of the grades array that has the lowest numeric score. Demonstrate the new methods in a complete program. Below is the UML diagram.


CourseGrades
- grades[ ] : GradedActivity
- NUM_GRADES : int = 4
+ CourseGrades()
+ setLab(aLab : GradedActivity) : void
+ setPassFailExam(aPassFailExam : PassFailExam) : void
+ setEssay(anEssay : Essay) : void
+ setFinalExam(aFinalExam : FinalExam) :
+ toString() : String
<< interface >>
Analyzable
+ getAverage() : double
+ getHighest() : GradedActivity
+ getLowest() : GradedActivity
LetterGrade
- score : double
+ setScore(s : double) : void
+ getScore() : double
+ getGrade() : char

Due date: March 24