Assignment #2
Class, Objects, Instance/Static Methods, Encapsulation

Requirements:

- Implement a class **GradeAnalyzer**. An object of this class accepts any number of grades, determines whether they are valid, then calculates their average, standard deviation, grade distribution (i.e., number of grades in each letter grade category), and the final GPA. Finally, it presents the results for all valid grades entered. For this assignment, use the grading scale as follows:

98 <= A+	88 <= B+ < 90	78 <= C+ < 80	60 <= D < 70
92 <= A < 98	82 <= B < 88	72 <= C < 78	F < 60
90 <= A- < 92	80 <= B- < 82	70 <= C- < 72	

  GPA for each grade:
  A+, A, A-, B+, B, B-, C+, C, C-, D and F (e.g. 4.3, 4, 3.7, 3.3, 3, 2.7, 2.3, 2, 1.7, 1, 0).
- You are required to implement a tester class **GradeAnalyzerTester**.

Class fields:

- Constants for each grade threshold, for example the minimum A+ = 98.
- Constants for the maximum and minimum permissible grades. For this assignment, the maximum = 100 and the minimum = 0.
- Magic numbers may not be used anywhere in your code! Use constants that follow naming conventions with appropriate visibilities.

Instance fields (i.e., attributes)

- Counter variables to accumulate the number of grades for each grade category. Need one for: A+, A, A-, B+, B, B-, C+, C, C-, D and F.
- Total number of grades entered
- Sum of all grades entered
- Sum of the square of each grade entered
- Average of all grades (rounded to the nearest integer)
- Standard deviation of the grades (rounded to the nearest integer)
- Total GPA of all grades entered
- Use sensible, human-readable names for all attributes with appropriate visibilities

Public interface

- Create a **constructor** for the **GradeAnalyzer** class
- isValidGrade(double aGrade):Boolean
  - Determines whether grade is valid (within acceptable grade range)
- addGrade(double aGrade):void
  - Calls isValidGrade(..) to determine whether to process or ignore the grade. Assuming the grade is valid……
    - Increments the appropriate letter grade counter
    - Increments the total number of grades entered and calculates the parameters necessary to calculate the average and standard deviation
- analyzeGrades():void
  - Calculates the average and standard deviation of scores
  - Calculates the GPA
  - Sets the appropriate attributes with the values calculated
- toString():String

    - Returns a properly formatted string of the grade information depending on the number of grades entered (must be same as sample output shown below!)
    - Appropriate getters & setters as needed
- **GradeAnalyzerTester** works as specified
- Every grade is passed to the GradeAnalyzer object whether valid or invalid!
- Use a loop to continue prompting until user wants to quit (Q or q)
- Calls analyzeGrades() and toString() to complete the processing & display results (see sample output below!)

Example

Please enter the grades and use Q to exit: 111, 100, 80, 60, 40, 20, 0, -1, q
Output:
You entered 6 valid grades from 0 to 100. Invalid grades are ignored!
The average = 50 with a standard deviation = 37
The grade distribution is:
A+ = 1
A = 0
A- = 0
B+ = 0
B = 0
B- = 1
C+ = 0
C = 0
C- = 0
D = 1
F = 3

Extra point (1%)

The GPA is 1.33

Submission: Your Eclipse project is named `yourStudentID_HW2`. Put all files in a folder and compressed it. Submit your assignment on eCourse under `HW2`. No other submissions will be graded and points will be deducted for late submission.

Academic dishonesty: You may not do work for another student nor may any student copy or plagiarize someone else's work. Severe penalties will be imposed on all parties involved.

Deadline: Thursday, December 3, 2020. (end of the day)