Assign-01a

Overloaded Functions

The Inputs

Write a program that uses overloaded functions to assess individual student grades. Each time through the program's main loop, it takes in a string that is in one of four forms:

- a single numerical grade as a floating point number that represents the student's final mark (ranging from 0.0 to 100.0)
- a letter grade that is one of the following: A+, A, B+, B, C+, C, D, F, I, Q, AU, DNA, I/P – again that represents the student's final grade
- from one to five numerical grades that are integers that represent the student's mark on each of five assignments worth 20% each, separated by spaces (the marks range from 0 to 100). Sample input might be :
    - 80 70 65 80 60 >>> which results in a final grade of 71.00 %
    - 80 90 50 >>> which results in a final grade of 44.00%
- a single character, 'X', that indicates that the user wants to exit the program
- Please note that this is **not** a menu-based program. That is, you are not to ask the user what type of grade they want to enter. You must take their input and determine what type of input they specified. This means you need to develop a function which intelligently parses their input to determine if they have entered a single double value, a single letter grade or from 1 to 5 integers. There is an executable that you can play with (in the *Parse-And-Validate.zip* file) that might help you figure it out.

The string will be analyzed to determine if the student passed, failed, or had a special situation. If the student passes or fails, display a line that gives the numerical value of the mark and an indication as to whether the student passed or failed. A mark of 54.50% or higher is required in order to pass. Your program's output should look like this:

```
Student achieved ###.## % which is a PASS condition.
Student achieved ###.## % which is a FAIL condition.
```
or

Important Notes:

- The two lines above will be the only allowed output from your program with the exception of
    - Special Situations (as noted below)
    - Any error messages that you need to output from your program
- Also note that in all input cases (the final grade, the letter grade and the assignment marks) – the program will output the grade as a floating point number (up to 2 decimal places)
- When testing your program I will only include valid input (i.e. a floating point number, an acceptable alpha-only input and a set of 1 to 5 integer values). Where
    - an acceptable floating point number may be "`60.5`" or "`12500.6`" or "`-12.76`"
    - an acceptable alpha-only input may be "`A+`", "`I/P`" or "`Hello`"
    - an acceptable set of integers may be "`1 2 3`", "`673 -23 70 30 20`"

Assign-01a

Overloaded Functions

- This means that I will not test with data like the following. It also means that you do not have to check for input given like this
    - I will not test with data like “6A.4”, “10 20 30 40 50 60 70”, “a+”, “50 A+ 30 45.2”, …
- In no input’s case will there be any trailing spaces so your solution should not depend on having trailing spaces
- If you see the need to create and add more functions to the solution – to properly modularize your solution design feel free to do so. A goal of every assignment, besides satisfying the requirements – is good design. As we discussed in SEF last term, please make sure that your new functions exist in the appropriate source module(s).

Special situations are denoted by the letter grades as follows:

Letter Grade	**Meaning**
I	Incomplete
Q	Withdrawal after drop/refund date
AU	Audit
DNA	Did Not Attend
I/P	In Process

If there is a special situation, then the output would show the information as follows:

```
Student has Special Situation :  AU (Audit Condition)
```

For display purposes, the numerical equivalents for the letter grades are:

Letter Grade	Numerical Equivalents
A+	95
A	85
B+	77
B	72
C+	67
C	62
D	57
F	50

Assign-01a

Overloaded Functions

What to Do

You must create three overloaded functions called assessGrade(). They will differ in their parameter lists:

- One will take a single parameter (char *) containing a letter grade or special situation.
- One will take a single parameter (double) containing the final mark.
- One will take five integer parameters containing assignment marks (or an array of five integers).

Review Module-01 for an understanding of how overloaded functions rely on each other (remember the "worker bee") and how to design a solution with overloaded functions. This program is to be written in C, but the source file extensions need to be `.cpp` when you save them (in order to invoke the C++ compiler in VS).

You will create and submit **only the following files**:

• assign1.cpp	contains the main loop requesting the user's input and the call to parse and determine the type of user input
• assessGrade.cpp	contains the logic for the 3 overloaded functions
• assessGrade.h	contains the prototypes for the 3 overloaded functions as well as any other defines or constants you may need

Please ZIP up and submit these files to the appropriate eConestoga Dropbox by the due date and time.