Assignment 5: Nested If Statements

Due February 3.

This assignment is more exercise with conditional statements and code. It builds on the previous assignment in that you must now use else conditions and conditional blocks. The attached documents contain reminders to help you remember how to use if, else, blocks, and nested if statements.

The exercise is designed in a special way to show you how you should write large programs by adding a little bit at a time. Please do these steps one at a time. The practice you will follow is called “refactoring.”

Step 1. Write the code that prompts the user to enter an integer between 1 and 20 (including 1 and 20), reads the value using cin, and then prints the value that they entered in a statement that begins with "You entered a ". Save this version in a separate location so that you can submit it.

Step 2. Add more code or modify your code so that if the user enters a value that is less than 1 or greater than 20, it prints out "The value you entered is not between 1 and 20." Otherwise, and only if the value is between 1 and 20, it prints the statement as before. Save this version in a separate location so that you can submit it.

Step 3. Add more code or modify your code so that if the user enters the values 8, 11, or 18, the statment that is printed is "You entered an" instead of "You entered a". Save this version in a separate location so that you can submit it.

Step 4. Add more code or modify your code so that if the users something that is not a number, i.e. includes letters, your program does not print any of the above lines, but instead prints out "Idiot! Your input could not be read as a number. Get a life!" Save this version in a separate location so that you can submit it.

For this last part, you must know that when cin cannot find a value that matches the type of the variable, it does nothing. If you assign a strange value to the variable before the cin statement, it will still have that strange value after the cin statement.

```
int iVal = -999;
cin >> iVal;
if (-999 == iVal) ...
```

As mentioned in the previous assignment, you can also test to see if the previous cin failed to read a value by calling cin.fail(). For this step, you can use either approach. If you didn’t do the cin.fail() part last time, I suggest you use the -999 method described above. (It’s simpler.)

Include comments in your code to explain what each of the ‘if’ statement is testing for.

Save each of the 4 versions separately, as a cpp file.
Zip the 4 cpp files in a single zip file and submit the 4 cpp files.
Do your work separately. Do not submit the same work as another student.
Do your work separately. Do not submit the same work as another student.