WICHITA STATE UNIVERSITY

CS 311: Object Oriented Programming
Spring 2018 – Assignment 1

For this assignment, in the spirit of wheat harvest, write a program to implement and test a new class named *Ticket*. Tickets are issued to track and report the loads of wheat delivered to the local grain elevator for short or long term storage.

When the grain truck containing wheat comes to a stop on the elevator's scale, the truck (including the wheat) is weighed. In addition, a sample of the wheat is taken and tested for its quality which includes weight, moisture level, and amount of foreign material (debris). Once the truck is unloaded of its wheat, the empty truck returns to the scale and is weighed again. The difference between these weights represents the net weight of the wheat that was in the truck. Using this net weight along with the wheat sample's test weight, moisture level, and foreign material, an overall number of net bushels can then be calculated using the following formula:

Net Weight (lbs.) = Gross Weight (lbs.) – Tare Weight (lbs.) (Tare is another way of saying empty)
Gross Bushels = Net Weight (lbs.) / 60 (lbs. per bushel)
Moisture Dockage (only if > 12%, otherwise 0) = Gross Bushels * (Moisture Level – 12.0%)
Foreign Material Dockage = Gross Bushels * Dockage %
Net Bushels = Gross Bushels – Moisture Dockage – Foreign Material Dockage

Use the following criteria for writing your program:

- ✓ Create a program named *WheatHarvest* (source file named WheatHarvest.cpp) to include the *main()* function along with additional helper functions required to obtain input, perform some processing logic, and show output results. However, most processing logic related to the ticket must be implemented by the class. **DO NOT perform any user input/output in a class.**

- ✓ Create a class named *Ticket* (header file named Ticket.h, source file named Ticket.cpp) to encapsulate/provide details about a single ticket issued for a load of wheat. The class should include the following:
  - Member variable for ticket number
  - Member variables for gross and tare weights
  - Member variables for moisture level and foreign material (both expressed as percentages)
  - Constant variable for average test weight per bushel (60 lbs / bushel)
  - Constant variable for ideal moisture level (12%)
  - Accessor/Mutator functions to get/set member variables as needed
  - Additional functions to provide logic and information to caller to promote encapsulation

To test the *Ticket* class, write a program to prompt for the ticket number, gross and tare weights, moisture level, and foreign material. For example: Prompt the user for a ticket number 101300A, a gross weight of 33,180 lbs., a tare weight 10,780 lbs., moisture level of 14%, and foreign material of 0.75%. Use class mutator functions to initialize member variables based on user input. Use class accessor functions to retrieve values of member variables and use additional functions to compute and return net weight, gross bushels, moisture dockage, foreign material dockage, and net bushels. Finally, output the ticket number, gross weight (lbs.), net weight (lbs.), gross bushels, moisture dockage (in bushels, but also show %), foreign material dockage (in bushels, but also show %), and resulting net bushels utilizing class member accessor and helper functions.

Example Input:

Ticket number: 101300A
Gross weight (lbs) 33180
Tare weight (lbs): 10780
Moisture level (%): 14.0
Foreign material (%): 0.75

Example Output (based on input):

Ticket 101300A:

33180	Gross Weight
10780	Tare Weight
22400	Net Weight
373.33	Gross Bushels
7.47	Moisture Level (14%)
2.80	Foreign Material (0.75%)
363.07	Net Bushels

Requirements and Submission:

- Due date: Saturday, January 20, 11:59pm
- Include the following header information (comment lines) at the beginning of each source/header file (before the #include directives) submitted. Notice each comment line begins with two slashes. C++ recognizes these as the start of single line comment. If you prefer, you may use block commenting.

```
// File Name: WheatHarvest.cpp
// Author: Firstname Lastname
// Student ID: ********
// Assignment Number: ##
```

Such as:

```
// File Name: WheatHarvest.cpp
// Author: Joe Shobe
// Student ID: xxxxxxxx
// Assignment Number: 1
```

- Grades will be based on:
  - ✓ User-friendly interactive input, output, and error messages
  - ✓ Programming style: Proper use of control statements, standard C++ techniques and naming conventions, and the conciseness and readability of the code
  - ✓ Accuracy: Does the program calculate and output correct data
  - ✓ Appropriate and descriptive comments
- Use of solutions found on the web, in the books, or from other students is <u>not</u> permitted. It must be your own work.
- Review the section *Submitting Assignments* in the Linux Server PDF for more information on how to turn in your assignment.
- No submissions will be accepted by e-mail, CD, DVD, or other independent storage or electronic media, unless approved by the instructor.