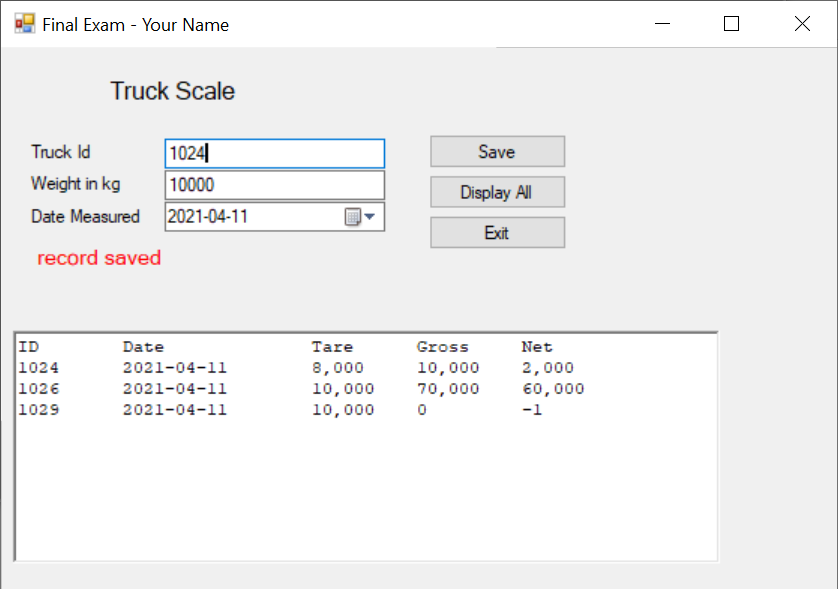
# PROG1815 (Programming Concepts II) – Final Exam

Please read the following instruction carefully before beginning the exam

* ~~The Final Exam folder on eConestoga is available on April 29~~~~th~~ ~~from 1:00 PM until 3:45 PM EDT~~
  + ~~Submissions will not be accepted after that time, including email.~~
  + ~~Zip and upload your solution periodically, and plan to finish with several minutes to spare.~~
    - ~~Your latest submission will be marked~~
    - ~~If you attempt to upload in the last few minutes and are unsuccessful, you will appreciate the fact you took a minute to upload 15 minutes earlier.~~
* ~~Using Visual Studio, create a windows project called XXFinalExam, where XX are your initials.~~
* ~~Create a form with your name on the form’s caption as shown.~~
  + ~~The display area at the bottom is a RichTextBox or a DataGidView~~



## ~~XXTruck Class~~

~~Create a class to manage a file of truck weight records~~. Methods are instance methods, unless otherwise stated.

1. This class has the following properties:
   1. TruckId – integer, greater than zero, unique
   2. DateMeasured – DateTime, formatted 2021-04-26 when displayed, not in the future
   3. ~~TareWeight, GrossWeight, NetWeight – integers, see edits.~~
2. ~~The class has just the default constructor:~~
   1. Initialises tare and gross weights to zero, NetWeight to -1 (signifying an incomplete record), and today’s date.
3. ~~Class-level Get methods:~~
   1. GetTrucks returns a collection of all Truck objects on file, ordered by TruckId
   2. ~~GetByTruckId accepts an Integer truckId and returns the Truck object with that ID, null if not on file.~~
4. CRUD methods:
   1. Save
      1. ~~This runs the edits and, if the object passes, decides which method processes it~~
      2. ~~If the TruckId is not on file, call the object’s Add.~~
      3. ~~If the TruckId is on file, but the NetWeight is more than -1, we’re starting a new measure for this truck, so call the object’s Update method.~~
      4. ~~If the TruckId is on file and the NetWeight is -1, we’re completing an existing record:~~
         1. ~~Put the greater of the two TareWeights into the current object’s GrossWeight~~
         2. ~~Put the lesser of the two TareWeights into the current object’s TareWeight~~
         3. ~~Compute the current object’s NetWeight as the difference between gross and tare~~
         4. ~~Call the current object’s Update method.~~
   2. ~~Add method:~~ 
      1. ~~Add the current object to the file as a single delimited line.~~
      2. ~~Catch any exception and display its message along with a lead-in text identifying the action attempted.~~
   3. ~~Update method:~~ 
      1. ~~Find the record on file with the same TruckId and replace it with the current object.~~
      2. ~~Catch any exception and display its message along with a lead-in text identifying the action attempted.~~
5. ~~Edit method has no parameters and a void return. All edit errors are thrown together as a single Exception, delimited to display one line per error.~~
   1. ~~The TruckId must be greater than zero.~~
   2. ~~The Date Measured must not be in the future.~~
   3. ~~The TareWeight is used to pass a new weight to the file. This must be greater than zero.~~
   4. ~~Force NetWeight to -1.~~
6. ~~ToString override method has no parameters and returns a string~~
   1. ~~Converts the object’s properties into a single string, with a delimiter between each field.~~
7. ~~Class-level Parse method accepts a string and returns a Truck object~~
   1. ~~Instantiates a Truck object, splits the string on its delimiters and populates the properties of the object.~~
8. ~~Class-level ConfirmFile, no parameters, void return~~
   1. ~~Called at the start of each Get method and the Save method to confirm the file exists or to create it if it doesn’t.~~

## The Truck Weight Maintenance Form

1. ~~The user fills in the three inputs: TruckId, weight and date measured.~~
2. ~~Save~~
   1. Instantiates an object of the Truck class
   2. Loads the truckId and date measured from the corresponding input fields.
   3. Puts the weight from the input field into TareWeight
   4. Calls the object’s Save method
   5. ~~Catches any exception and displays its message to the user.~~
3. Display
   1. Gets a collection of all truck records on file from the truck class’ GetTrucks method.
   2. Clears the display area.
   3. Loads them to the display area, one line per record, fields aligned with the headings and formatted as in the example
      1. **🡪 Note: if using a rich text box, change its font to Courier New … it makes alignments MUCH easier because “ “and “i” are the same width as “M” and “W”.**
4. ~~Exit~~
   1. ~~Closes the form.~~

