CSCI 470/502 **Assignment 2 – Classes and Objects** **Summer 2020**
50 points

Invoice Class

Create a class nameded `Invoice` that a hardware store might use to represent an invoice for an item sold at the store. An `Invoice` should include four pieces of information as instance variables – a part number (type `String`), a part description (type `String`), a quantity of the item being purchased (type `int`) and a price per item (`double`).

The class you create should has two constructors. One that initializes the four instance variables and another that takes no arguments. Provide a *set* and *get* method for each instance variable. In addition, provide a method named `getInvoiceAmount` that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0.

Write a test app name `InvoiceTest` that demonstrates class `Invoice`'s capabilities. `InvoiceTest` should declare five different invoice instances with different hardware store items. For example, hammer, phillips head screwdriver, light switch, cordless drill and carpenter's square.

You are going to be hardcoding the values for the instance variables for each of the five objects. It is not very realistic but it is okay to do this at this point.

Then, use the get methods to print each invoice, one after the other, as follows:

```
Invoice #1

        Part No.:  AB-23-4312

      Item Desc.:  Cordless Drill

        Quantity:  10

      Item Price:  189.00

Invoice Subtotal:       $1,890.00

*******************************

Invoice #2
```

and so on.

Note that, in the above, the item price for the cordless drill would have been set to 189.00.

Also, note that the `DecimalFormat` class should be used to place a floating dollar sign and commas in the output subtotal. The code and technique to do this can be found beginning on p. 8 of lecture notes 2. The maximum subtotal will be $99,999,999.99. This amount will help you determine your edit patter for DecimalFormat's method format.

Be sure to add the required doc box to the top of both classes and submit both .java files on Blackboard before the due date and time.