CST8110 - Introduction to Programming
Assignment #2 - Using interacting classes - Composition

DUE: Friday October 30 at 6pm SHARP – see submission instructions below. Late submissions receive a grade of 0.

Problem Description:
Write and test the program code in Java to solve the following problem:

- This problem will simulate producing a very basic invoice for a Telecom company (ie a company that provides **simple** cell phone service). Devise a name for your company, and use it in the first line of display from the program. (see example)

- The invoice will only handle one month bill for one client. We will call this class **Invoice**.

- The program should ask for the total ***number of daytime minutes*** of cell service to be purchased for the month (evening/weekends and texting are free for now).

- The program should calculate the ***invoice amount*** – which will be a flat rate of $5.00 per month plus the ***number of minutes purchased*** in the month multiplied by 2 cent per minute.

- The program should calculate and display the ***sales tax*** (13% of the ***invoice amount***) and the ***total amount due*** (ie ***invoice amount + sales tax***).

- Your code needs to format the output - specifically money amounts to two decimal places, etc.. Use either the **DecimalFormat** class or the **printf()** statement.

- You will use the **OurDate** class from Lab3 to handle the date functionality for this invoice – don't use the default Java class for Date. NOTE – we will use this class in a totally different program in another assignment (which will illustrate the advantage to using classes – reusability of code). Date data will consist of day, month and year. The class does NOT need to handle leap years. We will then use two objects of this **OurDate** class in our Invoice class – ***today's date and invoice date***.

- Your program should prompt the user to enter ***today's date*** and the ***invoice date*** then the invoice should display the number of days the invoice has been due to be paid. If the number of days overdue is over 60, you should add 10% interest to the bill (calculated AFTER the invoice amount has been calculated – but BEFORE taxes are included). Then note that the taxes is based on the bill amount WITHOUT the taxes – ie you do not pay tax on the interest.

- Your method main should be in class called **Assignment2.**

Suggested design for OurDate class (from lab3):

- ***Three private integer fields to store year, month and day***
- ***Methods:***
  - ***Constructor – default date of Jan 1, 1900***
  - ***Constructor – using three integers to set initial values for year, month and day***
  - ***setYearFromUser – prompt user to enter a year***
  - ***setMonthFromUser – prompt user to enter a month***
  - ***setDayFromUser – prompt user to enter a day***
  - ***displayDate – will display the date in yyyy/mm/dd format***
  - ***calcDays()– returns an int which is the number of days from Jan 1, 1900 – use 30 days per month for now, ∴ 360 days per year.***

Suggested design for Invoice class :

- ***Five private float or double fields to store minutesPurchased, invoiceAmount, interest, tax, totalAmount***
- ***Two private OurDate fields to store todayDate and invoiceDate***
- ***Methods:***
  - ***Constructor – default amounts to 0***
  - ***setMinutesFromUser – prompt user to enter cell service minutes***
  - ***setDatesFromUser – prompt user to enter today's date and invoice date***
  - ***calculateInvoice – calculate the invoice fields invoiceAmount, interest, tax and totalAmount***
  - ***displayInvoice – display the invoice***

**********************Opportunity for Bonus Marks*****************************

- Your code needs to worry if the user inputs an invalid value for the inputs (example negative number or outside an appropriate range). You do not need to worry about a user entering characters just yet – this would require a little bit more coding.
- Worth 5 extra marks but ***<u>ONLY</u>*** if all your other assignment features, as specified in the marking guide, are perfect – this assignment is potentially worth 25/20 marks

****

Sample Output #1: *(blue indicate user entered information)*

```
Welcome to the Crane Telecom Company Billing System
Enter the number of minutes you wish to purchase: 100
Enter today's date:
Enter a valid year: 2014
Enter the month: 3
Enter the day: 6
Enter invoice date:
Enter a valid year: 2014
Enter the month: 3
Enter the day: 2

Today date: 2014/3/6
Invoice date:2014/3/2

The number of days since invoice date: 4
Invoice amount : $7.00
Tax            : $0.91
Interest       : $0.00
TOTAL          : $7.91
```

Sample Output #2: *(blue indicate user entered information)*

```
Welcome to the Crane Telecom Company Billing System
Enter the number of minutes you wish to purchase: 1050
Enter today's date:
Enter a valid year: 2014
Enter the month: 3
Enter the day: 6
Enter invoice date:
Enter a valid year: 2014
Enter the month: 1
Enter the day: 1

Today date: 2014/3/6
Invoice date:2014/1/1

The number of days since invoice date: 65
Invoice amount : $26.00
Tax            : $3.38
Interest       : $2.60
TOTAL          : $31.98
```

Submission Requirements:

- You must create a .zip file that contains ONLY the following:
  - Your program code – .java files – (with your name, section, lab teacher listed in comments in the header of the program)
  - A document created with either Notepad, Wordpad or Word named Assign2.docx or Assign2.txt with your test plan – note this should contain your name, section and lab teacher listed at the top.
- The .zip file must have the following as it's name
  - Your last name, your first name, the word assign and the assign number …. Example RosenblumHowardAssign2.zip
- Submit the .zip file through the Assignment feature which has been enabled in the CST8110 Blackboard course. This should be directly under the Assignment description and will be published as soon as the due date for Assignment #2 has passed.
- Marks will be given for correct submission (ie marks will be deducted for incorrect submission!