CST8110 - Introduction to Programming
Assignment #4 – Arrays

DUE: Friday December 4th at 6pm SHARP – see submission instructions below. Late submissions receive a grade of 0.

Problem Description:

- This problem will create a monthly ledger for a user to keep track of monthly bills.
- The program will start by asking the user for an amount of monthly bills to pay.
- **Design of this solution will involve four classes, some may involve the use of previously written code:**

Class Design:

- **OurDate class – update your previously written OurDate class to include a toString() method. Be certain that you still have methods that will setDayFromUser(), setMonthFromUser() and setYearFromUser(). As always, explicitly write a default constructor.**

- **Invoice class – update your previously written Invoice class to include a name field (a String) and a toString() method. Be certain that you have methods that will** setDateFromUser(),setNameFromUser(), **and** setAmountFromUser(), **As always, explicitly write a default constructor.**

- **Ledger class – fields will include an array of Invoice references, and also an integer that will represent the number of invoices in the array. Two constructors are required, a default constructor (no input parameters) and also an overloaded constructor that receives an integer as an input parameter, and ultimately will be used in order to instantiate the Invoice array. Include methods getInvoiceInfo()** (invokes the setDateFromUser(),setNameFromUser(), **and** setAmountFromUser() methods from the Invoice class), printInvoiceInfo() (invokes the toString() method from the Invoice class, and a calculateMonthBills() (invokes a getAmount() method from the Invoice class that returns a double).

- **Assign4 class – this class will be the main() "driver" class. Create an object of Ledger class. Prompt the user for a number of invoices, and pass this integer as a formal parameter when instantiating the Ledger reference variable. Invoke the methods** getInvoiceInfo(),calculateMonthBills(), and printInvoiceInfo().

Sample Output : (blue indicates user entered information)

```
Enter the amount of monthly invoices: 3

Enter info for invoice number 0 :
Enter Company Name: Bell Canada
Enter bill amount: 47.10
Enter invoice due date:
Enter day: 1
Enter month: 12
Enter year: 2015

Enter info for invoice number 1 :
Enter Company Name: Enbridge
Enter bill amount: 55.38
Enter invoice due date:
Enter day: 5
```

```
Enter month: 12
Enter year: 2015

Enter info for invoice number 2 :
Enter Company Name: Hydro
Enter bill amount: 33.11
Enter invoice due date:
Enter day: 15
Enter month: 12
Enter year: 2015


Total monthly bills: 135.59

Bill Summary :
Bell Canada
47.10
1/12/2015

Enbridge
55.38
5/12/2015

Hydro
33.11
15/12/2015
```

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 each class)
  - A document created using Word, Notepad, or Wordpad named Assign4.docx or Assign4.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   CraneLindaAssign4.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.
- Marks will be given for correct submission (ie marks will be deducted for incorrect submission!
- You must use an array – does not use an Array List – marks will be deducted!

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

- Instead of prompting the user to "Enter the amount of monthly invoices" at the beginning of your code, define some upper limit that the amount of elements in the Invoice array is permitted to be. Use a repetition loop to ask the user if they need to add another Invoice to the array. All other functionality of the assignment, including the summing of the invoice amounts and the printing out a summary of the month's invoices must stand.
- Worth 5 extra marks but **ONLY** if all your other assignment features, as specified in the marking guide, are perfect – this assignment is potentially worth 25/20 marks

****