Georgian

COMP 1030

Assignment #7

Arrays

Introduction

During this assignment you will build two java classes. The first class will contain the required state and behaviour for the object, **<u>but NO main method</u>**. The second class will contain simply the main method , a line to instantiate a new object based upon the first class and a few lines to exercise the functionality of the first class.

When writing your code, keep these guidelines in mind:

- Start each class with a full javadoc comment header. The first line of that comment should be the purpose of the class.
- Provide a comment for **<u>each</u>** section of code.
- Follow the layout for your class as illustrated below:

    Javadoc comment header
    Import statements (if required)
    Class declaration
      State (instance variables/data)
      Constructor(s) (if required)
      Behaviour(s) (method(s))
    Close class declaration

- Use whitespace and indentations to make your code more readable and easy to debug.
- Be sure to clearly understand your work – do not simply copy code from someone else.

Instructions:

- Create a class that acts as a blueprint for a **BankTransaction** record.
    - This class should have the following state:
        - boolean **canadianFunds**
        - double **exchangeRate**
        - int **transactionNumber**
        - long **transactionReferenceNumber**
    - This class should have the following behaviour:
        - **A constructor** that takes four arguments.
        - **A public <u>class</u> variable** to keep track of the number of objects instantiated.
        - Appropriate **setters and getters** for all state

- Create a second class called **BankTransactionRecordTestHarness** to test the first class:
    - **Instantiate** 500 Bank Transaction Records.
    - **Each record should contain random data** for all state(use the java.util.Random class to accomplish this)
    - **Store** the reference for each object in an array.
    - **Print** out the state of each object along with the unique object number based upon the appropriate class variable, on separate lines followed by a blank line followed by a line of dashes (----------------------------------).
    - **Ask the user** if they wish to complete phase two of the test. If they say no end the program. If they say yes, set the state of each object to the same default value using the appropriate setters (you choose the value).
    - **Print** out the state of each object along with the unique object number based upon the appropriate class variable on separate lines followed by a blank line followed by a line of dashes (----------------------------------).

When writing your code, keep these guidelines in mind to ensure a maximum grade:

- The instructor reserves the right to schedule an oral review of any submission; if the student does not attend the oral review or cannot explain the code they have submitted they will receive a grade of zero.
- All naming conventions must be properly used in the code.
- All tabbing conventions must be properly used in the code.
- All variable and method names must be descriptive, single letter variables are not acceptable.
- The solution must be arrived at using constructs from the lectures up to and including the lecture to which this lab is associated.

- Code submitted must be written and understood by you.
- Submit a single PDF document in which the code is simply pasted into the document as text and the screenshots show the output.

Submission Instructions

Submit your completed work by uploading a **single PDF to BlackBoard**.

Your submission should contain a single copy of the entire code (**not** a screen shot of the code)

A screen shot (if appropriate) showing the output for each challenge.

Use only constructs that we have studied to date to solve this problem.

This is an individual assignment, “taking help” from another person is not permitted, all code must be authored by you, designed by you, and thought up by you.

Here are some tips to help you to complete this assignment:

- Read/listen to the lecture slides
- Reach out to your study group
- Email the instructor (Generally speaking industry will expect that you solve problems on your own, you are therefore encouraged to try your best to solve your problems yourself before reaching out to the instructor as his time is limited with approximately 200 students to coach.)

Java™
PROGRAMMING