Centennial College

COMP 228: Java Programming
Lab Assignment #1 – Java Classes

Due Date: <u>Mid-night (11.59 pm) Sunday Week02 -31st Jan</u>   Marks/Weightage: 30/5%

Purpose: The purpose of this Lab assignment is to:
- To create a console app using Java classes

IDE: eclipse enterprise Java Developer edition – 2020-06/2020-09
References: Learning materials for week 1,2 and textbook, and other references (if any)
This material provides the necessary information you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This lab should be completed individually by all the students.

Step1: At the start of eclipse, you must name your Eclipse workspace according to the following rule:
    YourFullName_COMP228SectionNumber_Lab01
    For Example: **JohnSmith_COMP228Sec002_Lab01 (if your section is Sec002)**

Step 2: And after that, you must name your Eclipse project according to the following rule:
    YourFullName_COMP228SectionNumber
    For Example: **JohnSmith_COMP228Sec002**

Step 3: And your package must be named as follows:
    YourFullName_SectionNumber_Ex01
    For Example: **johnsmith_sec002_ex01**

<u>**Submission/Upload Instructions:**</u>

After you complete, run and test your code, you need to do the following:
    a)   Close the eclipse, go to your **workspace** folder which you created in Step 1.
    b)   Zip it up. You should get a zip file like this– **JohnSmith_COMP228Sec002_Lab01.zip. You should only be submitting it in the .zip file format (and Not .rar or any other format)**
    c)   Upload this zip file using the Lab01 assignment link in e-centennial.

Apply the naming conventions for variables, methods, classes, and packages:
    - *variable names* start with a *lowercase* character for the first word and uppercase for every other word
    - *classes* start with an *uppercase* character of every word

      - **packages** use only *lowercase* characters
      - *methods* start with a *lowercase* character for the first word and uppercase for every other word

Note: Late submissions are NOT accepted. Assignment Submission Link will disappear if you miss the deadline. And after that you will not be able to submit your assignment. And it will not be considered as e-mail attachment.

Exercise 1:

Write a Java console application **(using IDE: eclipse Developer/enterprise Edition: 2020-06)** that is used to keep records of singers and displays stored record.
Follow the following instructions to develop the application:

Create a class named **Singer** (Singer.java) with the following specifications:
   a) 5 instance variables that would store the following singer data (Use recommended variable naming conventions and appropriate data type for each instance variable):   *[5 marks]*
       1) Singer's id
       2) Singer's name
       3) Singer's address
       4) Date of birth
       5) Number of albums published
   b) Several **overloaded constructors** that would allow you to construct Singer object with no arguments, 1 argument, 2 arguments, 3 arguments, 4 arguments, and 5 arguments.
      [5 marks]

   c) Create **Setters** and **getters** for all the instance variables of class Singer. Make sure to have several setters that would allow you to set the values of individual instance variables of the singer object. Also create one setter that would allow you to set all the values of the instance variables at once. Create several getters that would allow you to get the current individual values of each instance variables of the Singer object.   *[10 marks]*

   d) Create the **driver class (SingerTest – SingerTest.java)** that would create one Singer (singer1) object with the help of the no argument constructor. Display the default values of the instance variables of this object singer1.
      [5 marks]
   e) Set the values of each instance variables with the help of setters. Display the values.
      [5 marks]

Evaluation:

Functionality	
Correct implementation of classes (instance variable declarations, constructors, getters and setters methods, etc.)	45%
Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results)	45%
Friendly input/output	10%
Total	100%