Lab Exercise # 1 [1]

Note: The purpose of this exercise is to remind you of the terminology we learnt in CS115.

Note: Exercise 1 consists of Part A and B. Students will submit the solutions after they finish both Part A & B as a single zip file on blackboard.

Note: Students are expected to use the java compiler and interpreter from command line (from a DOS window). If you don't know how to do that raise your hand and ask one of us to help you out with that. Keep in mind that you will need to do the same with your own computer (or laptop) therefore you should learn how to download the Java Compiler/Interpreter and install it on your machine (ask for help on that task if you don't know how to do it).

Note: Use Notepad ++ or other approved IDEs, to write the program and COMMAND LINE TO COMPILE IT AND RUN IT.

Objectives:

- Go over the creation of service classes AND Client classes.
- Static variables.
- Static versus non-static methods in a class.
- Review toString AND equals methods (Equality of objects of a class).
- Use Java compiler and interpreter command line.
- Use command line and keyboard input data.

REMEMBER THE RULES FOR SUBMISSION.

- **Students must try to complete this exercise during class on 01/22 and have it approved (demo it to a TA) no later than 01/29.**
  - **You can demo it to a TA during the class or during TA office hrs.**
- Must be shown to a TA or instructor and be reviewed for correctness before it gets submitted.
- Upon approval by TA, you can submit it on Blackboard as single zip file. Watch for a checkmark in the blackboard grade center next to your name.
- Lack of check mark will be interpreted as an unauthorized submission.

PART A:

DESCRIPTION OF PROGRAM

We need to create a category of data called **Drivers**. The type of data that we need to collect for each driver is:

[1] Adapted from materials provided by George Koutsogiannakis

- The last name of the driver.
- The age of the driver as a whole number.
- The license number as a 5 digit whole number.
- The years that the driver has been driving as a decimal number.

Write the service class for the category of data Driver. Follow the pattern given to fill in the proper code:

public class (add the name of the class)
{

//write the declarations and initializations for the data attributes using identifiers of your choice.

//make sure that you use the proper data types for each attribute, based on the description given above.

//in addition, assign a unique id number to each object of the service class automatically via the constructors.

//write the default constructor for the class based on the fact that a default driver is a driver whose name is "No name", the age is 16, the license number is 11111 and the number of years driving is 0.5.

//provide the code to allow for the automatic creation of id numbers for each object.

//write the non default constructor with arguments ONLY the 4 basic attributes described above (again you should be automatically generating the ID).

//Write the accessor methods for all class variables
// Write the mutator methods for all class variables except for the ID
//Write the toString method for the class. The String returned by the toString method must have //an explanation in front of each attribute i.e "The name of the driver is:"+……+"The age is:"+……..+"The license number is:"+…… etc.

//write the equals method. Two service class objects are equal only and only if the driver's name and the driver's license number are equal for both objects.

} // end of class the class.

Compile this class COMMAND LINE!

PART B:

Now write the class client class for Drivers that you wrote for Part A. Name the class appropriately.

This class has two methods: a helper method and the main method.
The helper method is a NON STATIC method

//Write a help method whose signature is:

public int licenseFee(Driver dr)

//This method calculates the fee for the driver's license. The fee is the product of the age times the drivers license divided by 10000 (ten thousands). Then the value 50 is added to the result.

//However, the fee cannot exceed 100 Therefore smallest of the two numbers is chosen as the fee (i.e. if the calculation above is 135.6 then the smallest number between 100 and //135 is chosen as the fee).

//**The method returns the fee value as an int data type**.

In the main method of this class write code to:

- Instantiate (create) two objects of the service class.
- Data for the first object should be provided command line. The data will be used in the non default constructor of the service class.
- The second object receives the data required by the non default constructor of the service class from the keyboard.
- Display on the screen the values of all the attributes for the first object.
- Display on the screen the value of the license fee for that object.
- Display on the screen the values of all the attributes of the second object.
- Display on the screen the license fee value for the second object.

  Next write the code to determine if the two objects are equal or not and also display the result with the phrase "Command Line object is equal to Keyboard Object", if the objects are equal , or "Command Line object is NOT equal to the keyboard object" if there are not equal.

Compile this class COMMAND LINE

Run the program COMMAND LINE.

DEMO this to the TA in CLASS. Do not leave without showing to your TA

AFTER you demo, upload the code as a zip file to blackboard.

SAMPLE OUTPUT

C:\CS116\SPRING2015\EXTRACREDIT\SolutionPracticeProgram1\Solution>java DriversClient George 19 1234 2.5
Please enter the name of the driver
George
Please enter the age of the driver

34
please enter the license number
1234
Please enter the years of driving
12.3
The name of the driver is: George The age is: 19 The license number is: 1234 The years of driving is: 2.5
The fee for the first object is 52
The name of the driver is: George The age is: 34 The license number is: 1234 The years of driving is: 12.3
The fee for the second object is 54
The command line object is equal to the keyboard object