**FOP 2 - Lab 6**

**PART A:**

1. Type and compile the examples from the lecture.

**PART B:**

1. Modify the constructor and the set methods from the Car class (created in the previous lab) so that the make and registration cannot take the values of empty strings, while the engine size should take a value within 0.1 and 7.0 range; if empty strings and double values outside that range are passed, the instance variables should take the values of “some make”, “some registration”, and 0.1, respectively. Ensure that you also modify the main method to test for all these cases.
2. Create a PetRescueCenter classthat does the following (this exercise is very similar to exercise 3 from the previous lab, but each one of you should have a different implementation since each one of you have defined different attributes in the Pet class, and, therefore, different constructors and methods):

* creates 5 Pet objects (using the constructor of the Pet class that you have defined in the previous lab);
* create a list and declare that it holds Pet object references;
* add the 5 Pet objects to the list using the add method from the ArrayList class;
* display the pets in the list to potential owners using a for loop and the get method from the ArrayList class;
* You have noticed 2mistakes in relation to 2 different pets. Correct the mistakes using the set methods that you have defined in the Pet class;
* Re-display the list;
* 3 of the pets have been taken by potential owners and you need to remove them from the list using the remove(int index) method from the ArrayList class;
* Re-display the list.

1. Write a small program that contains the following 5 static methods (ensure that you test these methods with various strings by calling them in the main method):

* Tokenises a string/sentence and returns the number of occurrences of the word *car,* irrespective of its case.
* Tokenises a string/sentence and prints only the words that end with *ly.*
* Takes in a string/sentence, replaces all the instances of “ ” with “\_”, and then prints that string.
* Takes in a string/sentence, finds out all the indices of characters “e”, places them into a list and returns that list.
* Takes in 2 strings/sentences and prints which one comes first from an alphabetical perspective; if the strings are the same print “either one can be first”.

For this exercise you can assume that the strings/sentences contain only words delimited by a space, and it contains no punctuation, so you can tokenise the sentence using the space/white spaces only.

**Make sure that your code is indented and commented appropriately.**

**Save all your exercises, put them into a zip folder with your student number and submit it using the moodle upload facility for lab 6 by next Monday, 2.00 pm.**