Programming Languages 2
Lesson 9 – Test 2 Practice

1. Create a new NetBeans Java project. Set the name of the project to your given name. The name of the main class is `RetakeMain`. (2p)
2. Create a new class in the project. The name of this class is Car. *(Pay attention to place the class into the same package as the Main class.)* There are three attributes of this class: The brand is a String holding the name of the producer (e.g. Ford, Toyota, etc.). The age is an `int` variable holding the age of the car. The ID is a String that holds the license plate number of the car. *(It may contain numbers and alphabetic characters as well.)*. The attributes are hidden from the outside. (3p)
3. Create getter and setter methods for all the above attributes. (1p)
4. Create a custom Exception **in the same package** as the previous classes. The name of the exception is InvalidAgeException. Modify the setAge method of the Car class so that it throws InvalidAgeException if the user tries to set a negative number as the age of the car. (3p)
5. Override the toString method of the Car class. The method return a String like the following:

   ```
   brand - age: ID
   ```

   (e.g: Ford - 10: OMG-973)
   Take care to return a String **<u>exactly</u>** in this format. (3p)
6. Override the equals method of the Car. Two cars are equal if all their attributes are the same. (4p)
7. Add the following interface to the project (to the same package as the main class) and make the Car class implement it. (3p)

   ```
   public interface printable{
      public void print(); //prints out the object to the screen
                           //using its toString() method
    }//printable
   ```

8. Create a class named Garage. This class can hold Cars in inside. Make it implement the printable interface.
9. In the Main class, in the main method create a `Garage` that can hold Car objects. Create a car object with the following attributes: {brand: Suzuki; age: 13; license plate number: `LOL-999`}. Put this object into the `Garage`. The main function must not throw `InvalidAgeException`. (4p)
10. Read an `int` number from the keyboard. Read data from the keyboard for as many new Cars, as this number. (e.g. if the user types 4, read in data for 4 cars). If an exception appears, return from the main method. (5p)
11. Add each new Car to the `Garage`. (1p)
12. Print out the elements of the `Garage` using the `print()` method of the `it`. (1p)