Laboratory 3. Defining Class and Creating Instances

(Due: 3/23, Wed, 6pm)

Spring 2022
Soo Dong Kim

1. Write a Java program that implements the following class, *Car* for *Car Rental System*.

Car
int carID; **StatusType status;** **Date datePurchased;** **int mileage;**
Car (Date d, int m); **void setMileage (int x);** **void setStatus (StatusType s);** **public void printInfo( );**

- *CarID* is a 4-digit number identification number for cars, i.e., the range of 1,000 ~ 9,999.
  - Generate the *CarID* using a random number generator
    Check a potential conflict with existing *CarID* values. Upon a conflict, re-generate the *carID*.
  - Initialize the value of *CarID* inside the constructor. Since then, the value cannot be changed.
- *StatusType* is an enumerated datatype representing the status of each car. The datatype is defined with the values of (*available, checkedOut*, *inService*, *discarded*, and *sold*).
  - Initialize the value of *StatusType* with '*available*' inside the constructor.
- The 'd' and 'm' for the constructor are the initial values of *datePurchased* and *mileage* attributes respectively.
- Use *Date* and *SimpleDate* classes of Java to define *datePurchased*, that represents the date of car purchase. The usage is here;
  - import java.text.ParseException;
  - import java.text.SimpleDateFormat;
  - import java.util.Date;
  - public static void main(String args[]) throws ParseException {

```
// Define a format for printing a date
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
// Create a Date instance/object.
Date date = dateFormat.parse("2016-03-02");
// Print the date using the date format.
System.out.println(dateFormat.format(date));
}
```

2. Define a main( ) method performing the following tasks;

- Create 5 instances of *Car* with appropriate values for the input parameters to the constructor.
- Store the instances in an array *carArray*.
- Print the information of all the instances by invoking *printInfo( )*.
- Change the status of the first car instance by sending *setStatus(checkedOut)* message.
- Print the information of all the instances by invoking *printInfo( )*.
- Sort the cars in the array by their purchasing date in descending order, i.e. the most recently purchased car first.
- Print the information of all the instances by invoking *printInfo( )*.

3. Submission Guidelines

- Submit your solution on the web site; *myclass.ssu.ac.kr*
- Submit just **1 PDF file** containing the followings;
  - Java Source Code, *.java* file
  - Screenshot showing the program output
- Use this filename convention for your submission; **OOP.LAB.##.홍길동.pdf.**
  where ## is the laboratory number in 2 digits.
- No Plagiarism
  - The laboratory is an individual exercise. Do not copy others.
  - Submit your original work.

4. Grading Criteria (Total of 10 Points)

- Quality of Program (6)
  - Program Structure (4)
  - Exception Handling (1)
  - Header in the Source Program and Comments on Code (1)
- Accuracy of Output (4)
  - Correctness of Output Values (3)

    The output must be correct according to the problem specification.
  - Comprehensive Output Format (1)

    The output should be readable and comprehensive.
    Copy only the output part from the screen. (No Entire Screen)