CSD211 Lab4

1. [Preface](#_b0gerbgej72s)
2. [Objectives](#_hwd3jhtma1nn)
3. [Deliverables](#_ym7gdpu9vn58)
4. [Assignment](#_szyyosdlkbsr)

# Preface

* **Place a header at the beginning of each program that looks like this**

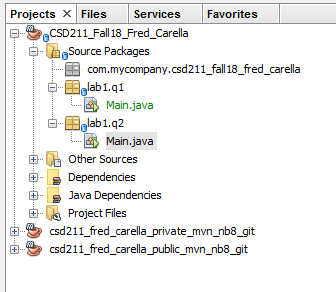
**/\*\***

**Author : Your Name**

**Date : the date**

**Description : describe what your program does.**

**\*\*/**

* **place your programs in packages.   
  put lab1 question 1 in package  
   lab2.q1  
  put lab1 question 2 in package  
   lab2.q2  
  etc…  
  If your program is in a class called Main.java then your packages should look like this…  
  **
* **commit and then push the programs to your git repo.**
* **place comments indicating the INPUT, PROCESS and OUTPUT sections of your program.**

# Objectives

* perform all the procedures outlined in the Assignment below

# Deliverables

* Complete the programming questions outlined below.  
  Then….
* in Netbeans, commit your files to your local repository.   
  **Do this every time you modify code.**
* in Netbeans, push all your code to your remote git repository (git/push).   
  **Do this frequently, after every coding session.**
* Answer any questions you are asked in a file called **Answers.docx**then upload Answers.docx to your dropbox.  
  **Do this when you complete answering the questions.**
* Upload a file called README to your LMS dropbox with the following information in it (a plain text file will work).   
    
  Name: your name  
  Date: the date  
  Repo: <https://gitlab.com>/<the url of your project. You get this from gitlab.com>|  
    
  **Do this when you’re done and the lab is ready for marking.**
* When you upload the README file to LMS dropbox, consider your lab handed in. Your teacher will be notified that the lab is submitted and that its ready for marking. Teacher marks it, and the mark automagically appears in your LMS Marks.  
    
  **Do not email me a message or a zip file of the project, after all we are in the 21’st century now :)**

# Assignment

For all questions, structure your code using the application skeleton:

Main.java

App.java

Put your code in the run method of the App.java class

In Main.java call "new App.run();"

1. Complete the Person exercise we discussed in class by doing the following
   1. Add constructors so that you can initialize a Person like this  
      // Joe Student is 23 years old and male.  
      Person p1=new Person("Joe", "Student", 23, 'M');  
        
      // George Student is a 24 year old male, is 182.88 cm tall and weighs 60kg.  
      Person p2=new Person("George", "Student", 24, 188.2, 60.0, 'M');  
        
      // Mary Teacher is female, 43 years old, is 170.00 cm tall and weighs 48.2 kg.  
      Person p3=new Person("Mary", "Teacher", 43, 170, 48.2, 'F');  
      System.out.println(p1);  
      System.out.println(p2);  
      System.out.println(p3);
   2. Add a toString method so it prints  
      Name : Joe Student  
      Age : 23  
      Height : 0.0cm  
      Weight :0.0kg  
      Gender : M  
        
      Name : George Student  
      Age : 24  
      Height : 188.2cm  
      Weight :0.0kg  
      Gender : M  
        
      Name : Mary Teacher  
      Age : 43  
      Height : 170.0cm  
      Weight :48.2kg  
      Gender : F
   3. Override the equals method. Write code that demonstrates whether or not 2 Person objects are equal.
2. Create a Car class with the following attributes.   
   Note use appropriate types.   
   Use the Person class from q1 for Owner.  
     
    Make  
    Model  
    Year  
    Owner  
    VIN  
   1. Create constructors so you can create cars like this:  
        
       Person shane=new Person("Shane", "Mollari");  
       Car ford=new Car("Ford", "Mustang", 2012);  
       Car chev=new Car("Chevrolet", "Camaro", 2012, shane);
   2. Add an appropriate toString method so you can print out each car. Make sure the full details of each owner get printed as well.
   3. Override the equals method. Write code that demonstrates whether or not 2 Car objects are equal.
3. Write a "used car lot" application that is implemented with the following features.  
   1. Create the following classes with the shown attributes and methods in the l4.q3 package.   
      (Note, I will elaborate further in the lecture on the methods and attributes for these classes.)  
        
      Car  
      ----------------  
      private String make;  
      private String model;  
      private int year;  
      private Person owner;  
      ----------------  
      Person  
      ----------------  
      private String firstName;  
      private String lastName;  
      ----------------  
      App.java  
      ----------------  
      private static Car[] cars=new Car[100];  
      private static int currentCar=0;  
      ----------------  
      Main (the driver)  
      ----------------  
      tbd  
      ----------------  
      1. Your app will implement a CRUD (Create, Update, Delete) menu for adding, editing and deleting cars from the used car lot.
         * When a car is sold, you will indicate the new owner of the vehicle by adding a person attribute to the car.
         * Cars will be stored in an array data structure (see UsedCarLot above).
         * Add cars to the end of the array.
         * When a car is deleted, the array should be `"re-packed", meaning, for example, if the first car is deleted, all the other cars will be shifted to to the left so to speak. There will be no "empty" car spots between cars.
         * Use "null" to indicate an empty car spot.
      2. The app should run in a loop. See lecture3.MenuLoop for an example of how to implement loops for this application. I will go over this in the lecture…  
         * 1. Add a car2.
           2. Sell a car
           3. List all cars
           4. Delete a car
           5. Edit a car
           6. Sell a car
           7. List sold cars and their owners.
           8. Enter make (-1 to quit):   
              -1
           9. Add a car2.
           10. Sell a car
           11. List all cars
           12. Delete a car
           13. Edit a car
           14. Sell a car
           15. List sold cars and their owners.
           16. Invalid entry, try again...