

# COMP1406 - Assignment #3

(Due: Mon. Feb 12th @ 12 noon)



In this assignment, you will simulate a police database that keeps track of a driver's history with respect to the number of times they were pulled over and ticketed for an infraction (i.e., a driving violation). The database will make use of **Arrays** so as to allow many drivers, vehicles and infractions to be stored together in lists. We will design a very small system and test it with only a few cars and owners.

## (1) The Driver, Vehicle & Infraction Classes

Define a class called **Driver** that defines the following attributes:

**license** - a String representing a driver's license ID (e.g. "L0678-67825-83940");

**name** - a String representing the name of the driver

**street** - a String representing the street name and number for this driver (e.g., "12 Elm St.");

**city** - a String representing the city in which this driver lives

**province** - a String representing the province in which this driver lives

Write the following in the **Driver** class:

- a proper constructor that takes 5 parameters for all 5 attributes in the order shown above
- a proper zero-parameter constructor that calls the first constructor with appropriate values
- a **toString()** method that returns a string in the following format:

```
"#L0678-67825-83940 John Doe living at 12 Elm St., Ottawa, ON"
```

Define a class called **Vehicle** that defines the following attributes:

**make** - a string representing the company that made the vehicle (e.g., "Pontiac", "Honda", etc..)

**model** - a string representing the model of the vehicle (e.g., "Grand Prix", "Civic", etc..)

**year** - an **int** representing the year of the vehicle (e.g., 2004 etc..)

**color** - a string representing the color of the vehicle (e.g., "red", "blue" etc..)

**plate** - a string representing the license plate of the vehicle (e.g., "X5T6Y8", "2FAST", etc..)

**owner** - a **Driver** object representing the owner of the vehicle (e.g., John Doe from part 1)

**reportedStolen** - a **boolean** indicating whether or not the vehicle has been reported as stolen

Write the following in the **Vehicle** class:

- a proper constructor that takes 5 parameters for the first 5 attributes in the order shown above (assume new vehicles are not stolen and there is no owner yet).
- a proper zero-parameter constructor that calls the first constructor with appropriate values
- a **toString()** method that returns a string in the following format:

```
"A blue 1998 Honda Civic with plate X5T6Y8"
```

Define a class called **Infraction** (i.e., an *infraction* is a violation) that defines the following attributes:

**amount** - a **float** indicating how much the fine was for this infraction

**description** - a **String** describing the infraction (e.g., "Not stopping for red light")

**dateIssued** - a **Date** representing the date and time at which the infraction occurred.

You must import **java.util.Date** in order for your code to compile.

**outstanding** - a **boolean** indicating whether or not the infraction was paid yet

**driver** - a **Driver** object representing the driver that received the infraction

Write the following in the **Infraction** class:

- a proper constructor that takes 3 parameters for the first 3 attributes in the order shown above
- a proper constructor that takes zero-parameters and calls the above constructor
- a **toString()** method that returns a string with the format below (use the **String.format()** method to format the amount to 2 decimal places and to display the date/time. The portion of the format string for displaying the date is "%tc". If the infraction has been paid, it should state [PAID IN FULL] as shown, otherwise it should state [OUTSTANDING]):

```
"$100.00 Infraction on Sun Jul 14 07:08:00 EDT 2002 [PAID IN FULL]"
```

- a method called **pay()** which pays the infraction (hint...use **outstanding** attribute)

---

## (2) The **PoliceDatabase** Class

Define a class called **PoliceDatabase** with the following instance variables:

**vehicles** – an **array** storing all vehicles in the database

**numVehicles** - an **int** that keeps track of how many vehicles there are in the database

**drivers** – an **array** storing all drivers (i.e., people who drive vehicles) in the database

**numDrivers** - an **int** that keeps track of how many drivers there are in the database

**infractions** – an **array** storing all infractions that have ever been given to drivers

**numInfractions** - an **int** that keeps track of how many infractions there are in the database

Write the following in the **PoliceDatabase** class:

- three static constants to represent the database size limits so that at most **2000 drivers**, at most **1000 vehicles** and at most **800 infractions** are allowed to be added to the database. Make sure that the rest of your code adheres to these exact limits.
- a zero-parameter constructor that initializes all the arrays properly
- a **registerDriver(Driver aDriver)** method which takes **aDriver** object as a parameter and then registers (i.e., remembers for later) the driver in the database ... assuming that the database is not full, otherwise the driver is not to be registered.
- a **registerVehicle(Vehicle aVehicle, String license)** method which takes **aVehicle** object as a parameter and the **license** ID of a driver who owns the vehicle (i.e., this is not the license plate of the vehicle) and then registers the vehicle in the database, making sure to also store the vehicle's owner properly ... assuming that the database is not full, otherwise the vehicle is not to be registered.

- a **unregisterVehicle(String plate)** method which takes a vehicle's **plate** number as a parameter and then removes the vehicle from the database without altering the order of the vehicles currently in the database and without leaving any gaps in the array. If the **plate** is not in the database, then nothing is to be done.
- a **reportStolen(String plate)** method which records that the vehicle with the given **plate** number has been stolen. If the **plate** is not in the database, then nothing is to be done.
- a **changeOwner(String plate, String license)** method which updates the database by changing the owner information for the vehicle with the given **plate** to the driver with the given **license ID** (i.e., this second parameter is not the license plate of the vehicle). If the **plate** or **license** is not in the database, then nothing is to be done.

### (3) Some Testing to Make Sure That It All Works So Far

Copy (cut and paste from the file, do not re-type) the **example()** method on the next page into your **PoliceDatabase** class. It is a method that builds and returns a **PoliceDatabase** object with some drivers and vehicles already registered. Re-compile to make sure that you have no errors.

```
public static PoliceDatabase example() { // Register all drivers and their vehicles
    PoliceDatabase pdb = new PoliceDatabase();

    pdb.registerDriver(new Driver("L1567-34323-84980", "Matt Adore",
        "1323 Kenaston St.", "Gloucester", "ON"));
    pdb.registerDriver(new Driver("L0453-65433-87655", "Bob B. Pins",
        "32 Rideau Rd.", "Greely", "ON"));
    pdb.registerDriver(new Driver("L2333-45645-54354", "Stan Dupp",
        "1355 Louis Lane", "Gloucester", "ON"));
    pdb.registerDriver(new Driver("L1234-35489-99837", "Ben Dover",
        "2348 Walkley Rd.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L8192-87498-27387", "Patty O'Lantern",
        "2338 Carling Ave.", "Nepean", "ON"));
    pdb.registerDriver(new Driver("L2325-45789-35647", "Ilene Dover",
        "287 Bank St.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L1213-92475-03984", "Patty O'Furniture",
        "200 St. Laurant Blvd.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L1948-87265-34782", "Jen Tull",
        "1654 Stonehenge Cres.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L0678-67825-83940", "Jim Class",
        "98 Oak Blvd.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L0122-43643-73286", "Mark Mywords",
        "3 Third St.", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L6987-34532-43334", "Bob Upandown",
        "434 Gatineau Way", "Hull", "QC"));
    pdb.registerDriver(new Driver("L3345-32390-23789", "Carrie Meehome",
        "123 Thurston Drive", "Kanata", "ON"));
    pdb.registerDriver(new Driver("L3545-45396-88983", "Sam Pull",
        "22 Colonel By Drive", "Ottawa", "ON"));
    pdb.registerDriver(new Driver("L1144-26783-58390", "Neil Down",
        "17 Murray St.", "Nepean", "ON"));
    pdb.registerDriver(new Driver("L5487-16576-38426", "Pete Reedish",
        "3445 Bronson Ave.", "Ottawa", "ON"));
    pdb.registerVehicle(new Vehicle("Honda", "Civic", 2015, "yellow", "W3EW4T"),
        "L0453-65433-87655");
    pdb.registerVehicle(new Vehicle("Pontiac", "Grand Prix", 2007, "dark green", "GO SENS"),
        "L0453-65433-87655");
    pdb.registerVehicle(new Vehicle("Mazda", "RX-8", 2004, "white", "OH YEAH"),
        "L2333-45645-54354");
    pdb.registerVehicle(new Vehicle("Nissan", "Altima", 2017, "bergundy", "Y6P907"),
        "L1234-35489-99837");
    pdb.registerVehicle(new Vehicle("Saturn", "Vue", 2002, "white", "9R6P2P"),
        "L2325-45789-35647");
    pdb.registerVehicle(new Vehicle("Honda", "Accord", 2018, "gray", "7U3H5E"),
        "L2325-45789-35647");
    pdb.registerVehicle(new Vehicle("Chrysler", "PT-Cruiser", 2006, "gold", "OLDIE"),
        "L2325-45789-35647");
}
```

```

pdb.registerVehicle(new Vehicle("Nissan", "Cube", 2010, "white", "5Y6K8V"),
    "L1948-87265-34782");
pdb.registerVehicle(new Vehicle("Porsche", "959", 1989, "silver", "CATCHME"),
    "L0678-67825-83940");
pdb.registerVehicle(new Vehicle("Kia", "Soul", 2018, "red", "J8JG2Z"),
    "L0122-43643-73286");
pdb.registerVehicle(new Vehicle("Porsche", "Cayenne", 2014, "black", "EXPNSV"),
    "L6987-34532-43334");
pdb.registerVehicle(new Vehicle("Nissan", "Murano", 2010, "silver", "Q2WF6H"),
    "L3345-32390-23789");
pdb.registerVehicle(new Vehicle("Honda", "Element", 2008, "black", "N7MB5C"),
    "L3545-45396-88983");
pdb.registerVehicle(new Vehicle("Toyota", "RAV-4", 2010, "green", "R3W5Y7"),
    "L3545-45396-88983");
pdb.registerVehicle(new Vehicle("Toyota", "Celica", 2006, "red", "FUNFUN"),
    "L5487-16576-38426");

return pdb;
}

```

Create the following class as a test program and make sure that it produces the proper output:

```

import java.util.GregorianCalendar;

public class PoliceDatabaseTestProgram {
    public static void main(String args[]) {
        PoliceDatabase pdb = PoliceDatabase.example();

        System.out.println("Here are the drivers:");
        for (int i=0; i<pdb.numDrivers; i++)
            System.out.println(pdb.drivers[i]);

        System.out.println("\nHere are the vehicles:");
        for (int i=0; i<pdb.numVehicles; i++)
            System.out.println(pdb.vehicles[i]);

        System.out.println("\nRemoving 3 vehicles (i.e., W3EW4T, FUNFUN and CATCHME) ...");
        pdb.unregisterVehicle("W3EW4T");
        pdb.unregisterVehicle("FUNFUN");
        pdb.unregisterVehicle("CATCHME");
        System.out.println("\nHere are the remaining vehicles:");
        for (int i=0; i<pdb.numVehicles; i++)
            System.out.println(pdb.vehicles[i]);

        System.out.println("\nHere is the owner of the Honda Accord:");
        System.out.println(pdb.vehicles[4].owner);
        System.out.println("Ilene Dover sells the car to Sam Pull, here is the owner now:");
        pdb.changeOwner("7U3H5E", "L3545-45396-88983");
        System.out.println(pdb.vehicles[4].owner);

        System.out.println("\nHere are the stolen cars at this time:");
        for (int i=0; i<pdb.numVehicles; i++)
            if (pdb.vehicles[i].reportedStolen)
                System.out.println(pdb.vehicles[i]);

        System.out.println("\nReport some cars as stolen...");
        pdb.reportStolen("OH YEAH");
        pdb.reportStolen("OLDIE");

        System.out.println("\nHere are the stolen cars now:");
        for (int i=0; i<pdb.numVehicles; i++)
            if (pdb.vehicles[i].reportedStolen)
                System.out.println(pdb.vehicles[i]);

        System.out.println("\nHere are some infractions:");
        Infraction i = new Infraction(75, "Illegal U-Turn",
            new GregorianCalendar(2017, 11, 14, 7, 8).getTime());
        i.pay();
        System.out.println(i);
        System.out.println(new Infraction(175, "Speeding in excess of 20km",
            new GregorianCalendar(2018, 1, 2, 14, 22).getTime()));
    }
}

```

## Here is the expected output:

### Here are the drivers:

#L1567-34323-84980 Matt Adore living at 1323 Kenaston St., Gloucester, ON  
#L0453-65433-87655 Bob B. Pins living at 32 Rideau Rd., Greely, ON  
#L2333-45645-54354 Stan Dupp living at 1355 Louis Lane, Gloucester, ON  
#L1234-35489-99837 Ben Dover living at 2348 Walkley Rd., Ottawa, ON  
#L8192-87498-27387 Patty O'Lantern living at 2338 Carling Ave., Nepean, ON  
#L2325-45789-35647 Ilene Dover living at 287 Bank St., Ottawa, ON  
#L1213-92475-03984 Patty O'Furniture living at 200 St. Laurant Blvd., Ottawa, ON  
#L1948-87265-34782 Jen Tull living at 1654 Stonehenge Cres., Ottawa, ON  
#L0678-67825-83940 Jim Class living at 98 Oak Blvd., Ottawa, ON  
#L0122-43643-73286 Mark Mywords living at 3 Third St., Ottawa, ON  
#L6987-34532-43334 Bob Upandown living at 434 Gatineau Way, Hull, QC  
#L3345-32390-23789 Carrie Meehome living at 123 Thurston Drive, Kanata, ON  
#L3545-45396-88983 Sam Pull living at 22 Colonel By Drive, Ottawa, ON  
#L1144-26783-58390 Neil Down living at 17 Murray St., Nepean, ON  
#L5487-16576-38426 Pete Reedish living at 3445 Bronson Ave., Ottawa, ON

### Here are the vehicles:

A yellow 2015 Honda Civic with plate W3EW4T  
A dark green 2007 Pontiac Grand Prix with plate GO SENS  
A white 2004 Mazda RX-8 with plate OH YEAH  
A bergundy 2017 Nissan Altima with plate Y6P907  
A white 2002 Saturn Vue with plate 9R6P2P  
A gray 2018 Honda Accord with plate 7U3H5E  
A gold 2006 Chrysler PT-Cruiser with plate OLDIE  
A white 2010 Nissan Cube with plate 5Y6K8V  
A silver 1989 Porsche 959 with plate CATCHME  
A red 2018 Kia Soul with plate J8JG2Z  
A black 2014 Porsche Cayenne with plate EXPNSV  
A silver 2010 Nissan Murano with plate Q2WF6H  
A black 2008 Honda Element with plate N7MB5C  
A green 2010 Toyota RAV-4 with plate R3W5Y7  
A red 2006 Toyota Celica with plate FUNFUN

Removing 3 vehicles (i.e., W3EW4T, FUNFUN and CATCHME) ...

### Here are the remaining vehicles:

A dark green 2007 Pontiac Grand Prix with plate GO SENS  
A white 2004 Mazda RX-8 with plate OH YEAH  
A bergundy 2017 Nissan Altima with plate Y6P907  
A white 2002 Saturn Vue with plate 9R6P2P  
A gray 2018 Honda Accord with plate 7U3H5E  
A gold 2006 Chrysler PT-Cruiser with plate OLDIE  
A white 2010 Nissan Cube with plate 5Y6K8V  
A red 2018 Kia Soul with plate J8JG2Z  
A black 2014 Porsche Cayenne with plate EXPNSV  
A silver 2010 Nissan Murano with plate Q2WF6H  
A black 2008 Honda Element with plate N7MB5C  
A green 2010 Toyota RAV-4 with plate R3W5Y7

### Here is the owner of the Honda Accord:

#L2325-45789-35647 Ilene Dover living at 287 Bank St., Ottawa, ON

Ilene Dover sells the car to Sam Pull, here is the owner now:

#L3545-45396-88983 Sam Pull living at 22 Colonel By Drive, Ottawa, ON

### Here are the stolen cars at this time:

Report some cars as stolen...

### Here are the stolen cars now:

A white 2004 Mazda RX-8 with plate OH YEAH  
A gold 2006 Chrysler PT-Cruiser with plate OLDIE

### Here are some infractions:

\$75.00 Infraction on Thu Dec 14 07:08:00 EST 2017 [PAID IN FULL]  
\$175.00 Infraction on Fri Feb 02 14:22:00 EST 2018 [OUTSTANDING]

## (4) More on the **PoliceDatabase** Class

Add the following methods to the **PoliceDatabase** class:

- **issueInfraction(String license, float amount, String description, Date date)** which causes the driver with the given license to receive an infraction for the given amount and description (i.e., reason) on the given date ... updating the database as necessary. This method **MUST** return the **Infraction** object created. If the number of infractions has reached its limit in the database, then nothing is to be done.
- **hasOutstandingInfractions(Driver d)** which returns a **boolean** with a value of **true** if the driver has any infractions that have not been paid yet, otherwise it should return **false**.
- **shouldStopVehicle(String plate)** which decides whether or not the police should pull this vehicle over. It should return a **boolean** with a value of **true** if and only if the vehicle has been **reportedStolen**, or if the vehicle's **owner** has at least one outstanding infraction (you **MUST** make use of the method **hasOutstandingInfractions()**). Otherwise, the method should return **false**.
- **showInfractionsFor(String license)** which displays a list of all infractions that the driver with this license ID has incurred showing (on successive lines) the driver's infractions ... and then finish with the total number of outstanding infractions. Here is the format you should use:

```
$100.00 Infraction on Sat Mar 03 11:55:00 EST 1990 [PAID IN FULL]
$250.00 Infraction on Tue Oct 06 09:22:00 EDT 1992 [PAID IN FULL]
$280.00 Infraction on Wed Aug 07 09:05:00 EDT 1996 [PAID IN FULL]
$300.00 Infraction on Fri Mar 03 12:15:00 EST 2000 [OUTSTANDING]
$350.00 Infraction on Mon Feb 01 02:04:00 EST 2010 [OUTSTANDING]
Total outstanding infractions = 2
```

- **cleanDrivers()** which returns an **Driver[ ]** of all drivers who have never had any infractions. The array should have a length corresponding exactly to the number of clean drivers (i.e., no nulls).
- **showInfractionReport()** which lists information about all drivers who have had at least one infraction. Each line in the list should show a driver's name (formatted to take 20 spaces ... using the **String.format()**) followed by the number of infractions that they received and the total amount for all infractions that they have already paid (formatted as shown) as follows:

```
Bob B. Pins: 1 infractions, total paid = $ 75.00
Ilene Dover: 4 infractions, total paid = $400.00
   Jim Class: 5 infractions, total paid = $630.00
Bob Upandown: 1 infractions, total paid = $ 0.00
... etc...
```

---

## (5) Final Testing

Define a class called **PoliceDatabaseTestProgram2** as shown on the next page. Cut and paste the code... do not re-type it. It would be a good idea to write blank methods for part 4 so that the test code compiles and runs. Then fill in the methods one at a time and test them.



```

import java.util.GregorianCalendar;
public class PoliceDatabaseTestProgram2 {
    public static void addInfractions(PoliceDatabase pdb) {
        pdb.issueInfraction("L0453-65433-87655", 75, "Illegal U-Turn",
            new GregorianCalendar(2010, 6, 14, 7, 8).getTime()).pay();
        pdb.issueInfraction("L2325-45789-35647", 175, "Speeding in excess of 20km",
            new GregorianCalendar(2011, 2, 2, 14, 22).getTime()).pay();
        pdb.issueInfraction("L2325-45789-35647", 75, "Illegal U-Turn",
            new GregorianCalendar(2012, 3, 22, 4, 34).getTime()).pay();
        pdb.issueInfraction("L2325-45789-35647", 150, "Wrong way up 1-way street",
            new GregorianCalendar(2014, 6, 14, 8, 2).getTime()).pay();
        pdb.issueInfraction("L2325-45789-35647", 500, "Running red light",
            new GregorianCalendar(2017, 9, 12, 5, 15).getTime());
        pdb.issueInfraction("L0678-67825-83940", 100, "Failure to signal",
            new GregorianCalendar(2008, 2, 3, 11, 55).getTime()).pay();
        pdb.issueInfraction("L0678-67825-83940", 250, "Speeding in excess of 30km",
            new GregorianCalendar(2010, 9, 6, 9, 22).getTime()).pay();
        pdb.issueInfraction("L0678-67825-83940", 280, "Speeding in excess of 30km",
            new GregorianCalendar(2015, 7, 7, 9, 5).getTime()).pay();
        pdb.issueInfraction("L0678-67825-83940", 300, "Speeding in excess of 30km",
            new GregorianCalendar(2017, 11, 3, 12, 15).getTime());
        pdb.issueInfraction("L0678-67825-83940", 350, "Speeding in excess of 30km",
            new GregorianCalendar(2018, 1, 1, 2, 4).getTime());
        pdb.issueInfraction("L6987-34532-43334", 500, "Running red light",
            new GregorianCalendar(2017, 7, 2, 22, 17).getTime());
        pdb.issueInfraction("L3545-45396-88983", 75, "Illegal U-Turn",
            new GregorianCalendar(2007, 3, 4, 20, 12).getTime()).pay();
        pdb.issueInfraction("L3545-45396-88983", 175, "Speeding in excess of 20km",
            new GregorianCalendar(2011, 5, 4, 23, 25).getTime()).pay();
        pdb.issueInfraction("L1144-26783-58390", 175, "Speeding in excess of 20km",
            new GregorianCalendar(2012, 10, 1, 19, 38).getTime());
        pdb.issueInfraction("GARBAGE ID", 100, "Loving Java too much",
            new GregorianCalendar(2017, 11, 24, 19, 38).getTime());
    }
    public static void main(String args[]) {
        // Make a new database and add some vehicles/owners etc..
        PoliceDatabase pdb = PoliceDatabase.example();
        addInfractions(pdb);

        System.out.println("Vehicles that should be stopped:");
        for (int i=0; i<pdb.numVehicles; i++)
            if (pdb.shouldStopVehicle(pdb.vehicles[i].plate))
                System.out.println(pdb.vehicles[i]);

        System.out.println("\nReport OH YEAH, W3EW4T and LUVJAVA as stolen:");
        pdb.reportStolen("OH YEAH");
        pdb.reportStolen("W3EW4T");
        pdb.reportStolen("LUVJAVA"); // This one won't work, but shouldn't crash

        System.out.println("\nVehicles that should be stopped:");
        for (int i=0; i<pdb.numVehicles; i++)
            if (pdb.shouldStopVehicle(pdb.vehicles[i].plate))
                System.out.println(pdb.vehicles[i]);

        System.out.println("\nHere are all the clean drivers:");
        for (Driver d: pdb.cleanDrivers())
            System.out.println(d);

        System.out.println("\nHere is the infraction report:");
        pdb.showInfractionReport();

        System.out.println("\nHere is the status of Jim Class:");
        pdb.showInfractionsFor("L0678-67825-83940");
    }
}

```

Here is the proper output for the above test case ... make sure that it works:

```
Vehicles that should be stopped:  
A white 2002 Saturn Vue with plate 9R6P2P  
A gray 2018 Honda Accord with plate 7U3H5E  
A gold 2006 Chrysler PT-Cruiser with plate OLDIE  
A silver 1989 Porsche 959 with plate CATCHME  
A black 2014 Porsche Cayenne with plate EXPNSV
```

Report OH YEAH, W3EW4T and LUVJAVA as stolen:

```
Vehicles that should be stopped:  
A yellow 2015 Honda Civic with plate W3EW4T  
A white 2004 Mazda RX-8 with plate OH YEAH  
A white 2002 Saturn Vue with plate 9R6P2P  
A gray 2018 Honda Accord with plate 7U3H5E  
A gold 2006 Chrysler PT-Cruiser with plate OLDIE  
A silver 1989 Porsche 959 with plate CATCHME  
A black 2014 Porsche Cayenne with plate EXPNSV
```

Here are all the clean drivers:

```
#L1567-34323-84980 Matt Adore living at 1323 Kenaston St., Gloucester, ON  
#L2333-45645-54354 Stan Dupp living at 1355 Louis Lane, Gloucester, ON  
#L1234-35489-99837 Ben Dover living at 2348 Walkley Rd., Ottawa, ON  
#L8192-87498-27387 Patty O'Lantern living at 2338 Carling Ave., Nepean, ON  
#L1213-92475-03984 Patty O'Furniture living at 200 St. Laurant Blvd., Ottawa, ON  
#L1948-87265-34782 Jen Tull living at 1654 Stonehenge Cres., Ottawa, ON  
#L0122-43643-73286 Mark Mywords living at 3 Third St., Ottawa, ON  
#L3345-32390-23789 Carrie Meehome living at 123 Thurston Drive, Kanata, ON  
#L5487-16576-38426 Pete Reedish living at 3445 Bronson Ave., Ottawa, ON
```

Here is the infraction report:

```
Bob B. Pins: 1 infractions, total paid = $ 75.00  
Ilene Dover: 4 infractions, total paid = $400.00  
Jim Class: 5 infractions, total paid = $630.00  
Bob Upandown: 1 infractions, total paid = $ 0.00  
Sam Pull: 2 infractions, total paid = $250.00  
Neil Down: 1 infractions, total paid = $ 0.00
```

Here is the status of Jim Class:

```
$100.00 Infraction on Mon Mar 03 11:55:00 EST 2008 [PAID IN FULL]  
$250.00 Infraction on Wed Oct 06 09:22:00 EDT 2010 [PAID IN FULL]  
$280.00 Infraction on Fri Aug 07 09:05:00 EDT 2015 [PAID IN FULL]  
$300.00 Infraction on Sun Dec 03 12:15:00 EST 2017 [OUTSTANDING]  
$350.00 Infraction on Thu Feb 01 02:04:00 EST 2018 [OUTSTANDING]  
Total outstanding infractions= 2
```

---

## IMPORTANT SUBMISSION INSTRUCTIONS:

Submit your **ZIPPED IntelliJ project file** as you did during the first tutorial for assignment 0.

- **YOU WILL LOSE MARKS IF YOU ATTEMPT TO USE ANY OTHER COMPRESSION FORMATS SUCH AS .RAR, .ARC, .TGZ, .JAR, .PKG, .PZIP.**
  - **If your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !**
  - **You WILL lose marks on this assignment if any of your files are missing. So, make sure that you hand in the correct files and version of your assignment. You will also lose marks if your code is not written neatly with proper indentation. See examples in the notes for proper style.**
-