EECS168/169-Lab10

Classes & Objects
Object Oriented Programming
DMV Program

DMV Program

Objectives:

- Designing classes
- Arrays of Objects
- File I/O
- Command line arguments
- Input validation

You ***must*** use these to implement your program

- Write a program that reads in Drivers License Records from **file** &
- then let the user interact with the data.
  - Validating all iterations with the user that involve obtaining an int
    - Menu selections
    - Ages
    - Drivers license numbers

DMV: Data File

<number of entries>

<first name> <last name> <age> <registered voter?> <drivers license number>

<first name> <last name> <age> <registered voter?> <drivers license number>

<first name> <last name> <age> <registered voter?> <drivers license number>

```
55
Francine Palau 23 N 730248
Kam Swindler 57 Y 468119
Migdalia Constable 21 N 493364
…
```

Requirements

Lots of files, variables, functions!

Classes: DriversLicenseRecord, DMV

Code the skeletal of files ***FIRST*** –

1. `DriversLicenseRecord.h & DriversLicenseRecord.cpp`

2. `DMV.h & DMV.cpp`

3. `main.cpp`

4. `makefile`

Requirements

1. DriversLicenseRecord.h & DriversLicenseRecord.cpp
2. DMV.h & DMV.cpp
3. main.cpp
4. makefile

Makefile Updated:

```
DriversLicenseRecord.o: DriversLicenseRecord.h DriversLicenseRecord.cpp
	g++ -std=c++11 -c DriversLicenseRecord.cpp

DMV.o: DMV.h DMV.cpp
	g++ -std=c++11 -c DMV.cpp

main.o: main.cpp DMV.h DriversLicenseRecord.h
	g++ -std=c++11 -c main.cpp

dmvprogram: main.o DriversLicenseRecord.o DMV.o
	g++ -std=c++11 main.o DriversLicenseRecord.o DMV.o -o dmvprogram
clean:
	rm *.o dmvprogram
```

DMV Program

```
/* Filename: DriversLicenseRecord.h */
class DriversLicenseRecord        {
    // Member variables
    string m_first_name;
    stirng m_last_name;
    int    m_age;
    bool   m_voter status;
    int    m_license number;

public:
    //getters and setters
    string get_first_name();
    void set_first_name(string first_name);
    ....
    ....
};
```

DMV Program

```
/* Filename: DriversLicenseRecord.cpp */
#include "DriversLicenseRecord.h"

//Define getters and setters
string DriversLicenseRecord::get_first_name()
{

}

void  DriversLicenseRecord::set_first_name(string first_name)
{

}
```

DMV Program

```
/* Filename: DMV.h */
class DMV       {
    // Member variables
    // ?

public:
    // Constructor
    DMV(string filename);


    ....
    ....

    // Interacting with the user
    // Print menus
    // Validate user input!
    ....
    ....

    //All I/O to the terminal
    ....
    ....

};
```

DMV Program

```
/* Filename: DMV.cpp */
#include "DMV.h"

// Constructor
DMV::DMV(string filename)
{
    //Reading and storing the drivers license records
    //Make sure the file exist!
}

//Define getters and setters
string DMV::get_some_variable()
{

}
```

DMV Program

```
/* Filename: main.cpp */
#include <iostream>
#include "DMV.h"
#include "DriversLicenseRecord.h"

using namespace std;

int main(int argc, char** argv)
{
    //Do a check to make sure we have the right number of command line arguments, exit if
    //there are not enough arguments
    std::string fileName =???; //get the file from argv
    DMV myDMV(fileName);
    myDMV.run();

}
```

DMV Program Options

```
Select an option:
1) Print all Drivers Info
2) Print all voters
3) Print drivers by last initial
4) Print drivers in age range
5) Quit
Enter your choice:
```

Option	Description

Print all Driver Info	Prints all drivers and all their information in the following format:
	`<last name>, <first name> (<age>): <drivers license number>`
	Example of a single entry (you'll print all entries)
	`Johnson, Larry (67): 301288`

Print all voters	Prints the driver information in the same format as the previous option, but only prints those registered to vote.
Print drivers by last initial	Asks the user for single character, and print the information for all drivers that have a last name starting with that letter (case-insensitive) OR print "No record found." if no drivers have a last name starting with that character
Print drivers in age range	Prompts the user for two ints that represent an age range. Print all drivers (following the same format as when printing 1 driver) within this age range
Quit	Exits the program.

169 Requirements

Add the following option to the menu:

- Register to vote
  - This option will prompt the user for a drivers license number
  - If the drivers license number exists and they are not registered to vote, you register the driver
  - If the driver doesn't exist, display "No record found"
  - If the driver is already registered, display "Driver currently registered"
- When your program exits, the input file (whatever the name maybe) should have it's records updated.