WEEK 4 EXERCISES

Please submit to the assignment folder on d2l by due date

Exercise 1: Car class (7 points)

Create a header file (car.h) with a class definition for a Car object.

A Car object will have three member variables:

- a make (cstring)
- a model(cstring)
- a year(int).

After creating the class definition, create a car.cpp file and implement the member functions in the class definition.

NOTE: You will need a 'getter' and a 'setter' for each member variable.

Download the carmain1.cpp file from d2l and test your class by putting all files inside one directory and compiling them with:

```
g++  *.cpp  -o prog
```

Requirements:

- All member variables are private.
- All member variables have a corresponding getter and setter member function. See carmain1.cpp for what these should be called.
- No string type in your class. Just C-Strings

There should be TWO files to submit: a car.cpp file and a car.h file. Put these in a directory called 'week3exer1'. Then make a tar.gz of the directory:

```
tar cvzf week3exer1.tar.gz week3exer1/
```

Use carmain1.cpp for testing as I will be using this file to test your submission (don't write your own).

```
-------------------------------
EXAMPLE OUTPUT #1
--------------------------------
Sedan before initialization:
Make: #��m#
Model:
Year: 0
Setting values for sedan:
Enter make: Honda
Enter model: Civic
Enter year: 2003
printing values for sedan:
Make: Honda
Model: Civic
Year: 2003
```

Exercise 2: Car Class Constructors (5 points)

Take your Car class and add two constructors. One constructor takes no parameters. It should set cstrings to empty strings and set numbers to 0.

```
    make[0] = '\0';
```

The other constructor should take make, model and year parameters and set the corresponding member variables to these values.

Download carmain2.cpp and test your class on this code.

Requirements

Same requirements as Exercise 1 except you need to have a default constructor and a parameterized constructor in your Car class.

```
-------------------------------
EXAMPLE OUTPUT #1
-------------------------------
Sedan before initialization:
Make:
Model:
Year: 0
Compact values
Make: Honda
Model: Civic
Year: 2017
Setting values for sedan:
Enter make: Ford
Enter model: Escort
Enter year: 2012
printing values for sedan:
Make: Ford
Model: Escort
Year: 2012
```

Exercise 3: Car Class make file (3 points)

Use the examples in the makefile slides to write a simple make file for Exercise 2. Remember you need to call the file 'Makefile' or 'makefile'

Now test it by deleting the program executable (and any .o files) from the directory and running the command:

make

this should go through the process of compiling and linking your files.

Requirements

Same exercise 2 except include a make file in your directory.