**CSCI 1411 – Lab 07 – Arrays, typedef, and structs**

**Goals:**

* Accessing variables in Arrays
* Use of typedef
* Definition and use of structs
* Accessing variables in arrays of structs
* Passing arrays into functions

**Development Environment:** (all students must use Visual Studios)

**Skills:** arrays, typedef, structs, arrays of structs

**Reading**: Chap 7, Chap 11

**Deliverables:** 1) This lab with 4 screen shots 2) lastnameFirstLab07.cpp

**Part I – Skills Practice (5 points)**

* Open a new project in Visual Studios. Call the project Lab07

You will be creating 4 files called lab07a.cpp, Car.h and CarFunctions.h, CarFunctions.cpp

In Car.h

Remember in your Car.h, you should remove the #pragma once, and add the duplicate guard:

**#ifndef CAR\_H**

**#define CAR\_H**

**#include <string>**

**using namespace std;**

**struct Car**

**{**

**string make;**

**string model;**

**int year;**

**};**

**#endif //CAR\_H**

* Now create a file called **CarFunctions.h**

**#ifndef CARFUNCTIONS\_H**

**#define CARFUNCTIONS\_H**

**#include "Car.h"**

**Car newCar(istream &in);**

**bool addCar(Car CarToAdd, Car carArray[], int &currentSize, const int MAXSIZE);**

**void listCars(ostream &out, Car carArray[], int &currentSize);**

**#endif**

* Now create a file called **CarFunctions.cpp**

**#include <iostream>**

**#include "Car.h"**

**#include "CarFunctions.h"**

**using namespace std;**

**Car newCar(istream &in)**

**{**

**Car car1;**

**cout << "Make: ";**

**in.ignore();**

**getline(in, car1.make);**

**cout << "\nModel: ";**

**getline(in,car1.model);**

**cout << "\nYear: ";**

**in >> car1.year;**

**cout << endl;**

**return car1;**

**}**

**bool addCar(Car carToAdd, Car carArray[], int &currentSize, const int MAXSIZE)**

**{**

**if (currentSize < MAXSIZE)**

**{**

**carArray[currentSize] = carToAdd;**

**currentSize++;**

**return true; //successful add**

**}**

**else**

**return false;**

**}**

**void listCars(ostream &out, Car carArray[], int &currentSize)**

**{**

**for (int i = 0; i < currentSize; i++)**

**{**

**out << "Car " << i + 1 << endl;**

**out << carArray[i].make << endl;**

**out << carArray[i].model << endl;**

**out << carArray[i].year << endl<<endl;**

**}**

**}**

* Finally, create a file called **lab07a.cpp**

**#include <iostream>**

**#include "Car.h"**

**#include "CarFunctions.h"**

**using namespace std;**

**int main()**

**{**

**const int MAXSIZE = 2;**

**int currentSize = 0;**

**Car car1;**

**Car carArray[MAXSIZE];**

**int menu;**

**bool done=false;**

**while (!done)**

**{**

**cout << "1. Add a new car to the array\n";**

**cout << "2. List out cars\n";**

**cout << "3. exit\n";**

**cin >> menu;**

**switch (menu)**

**{**

**case 1: car1 = newCar(cin);**

**addCar(car1, carArray, currentSize, MAXSIZE);**

**break;**

**case 2: listCars(cout, carArray, currentSize);**

**break;**

**case 3: exit(0);**

**break;**

**default: cout << "Number between 1 and 3\n";**

**}//switch**

**}//while**

**}//main**

* **Run** it adding two cars. Take a **screen shot** of the successful output and place it below. For a Windows 10 screen shot: Alt key + PrtSc key. Then Ctrl + V to paste. For Mac: Shift + Command + 4. You will not credit unless you have a successful screen shot with Your name in the output.

**Part II – Troubleshooting Common Problems (5 pts)**

Two common problems with functions include:

* Missing semi-colon after struct definition.
* Going past the end of the array

Go into Car.h and remove the semi-colon after the closing }

You should get an error when you build it.

* Take a **screen shot** of the error and place it below.
* Put the semi-colon back
* Now validate what the error looks like when you try to access the
* Go to CarFunctions.cpp
* Change the AddCar function line

**carArray[currentSize] = carToAdd;**

* to

**carArray[currentSize - 1] = carToAdd;**

Build it, Run it and add 3 records. Then you should get an error.

* Take a **screen shot** of the error and place it below.

Change back to currentSize -1.

Save and exit then,

**Part III - outline/pseudo-code/algorithms (5 pts)**

* Work with your lab partner to write an outline and comments in psuedocode to complete the program in Part IV. Use plain English for your outline.
* You should have one file broken into sections with comments. Each .h and .cpp should be in a separate section in the single lastnamefirstLab08.cpp. Every program you write should have the following block at the top in comments. Make sure to fill in the Name, Class, Description and Lab Partner at the top of the file. Ensure your status is accurate.

/\* Name:  
Class: CSCI 1411-00X  
Description: [fill in description]

Lab Partner:  
Status: successfully compiled and run on csegrid [if it doesn’t run or meet all of the requirements, list the actual status!/\*

**Part IV -Movie Listing. (10 pts)**

* Write a program that reads in data from a file and places the information in a struct named MovieData. The input file has between zero and 100 “records” each with the all of following items (in order, separated as noted below)

Title

ReleaseYear

RunningTime

Rating

Title might have spaces in them. RunningTime is in minutes. Rating is something similar to RottenTomatoes.com, so listed in a percent.

You will have a menu that continues until your press exit, and returns an error and goes back to the menu if not one of the choices. You can assume the user will only enter integers for the menu:

1. Read in Movies
2. Add Movies
3. List Movies
4. Exit

You will add to this menu of items for next weeks lab.

For ease of grading, we will have you place all of the code in a single file called yourlastnamefirstLab07.cpp. Your PAs for CSCI 1410 will require multiple files, but now that we know how to use them, combine the files. Your .cpp file should have the following format:

/\* Name:  
Class: CSCI 1411-00X  
Description: [fill in description]

Lab Partner:  
Status: successfully compiled and run on csegrid [if it doesn’t run or meet all of the requirements, list the actual status!/\*

//Function Prototypes

//Function Implementations

int main()

Since they are all together in one file you not have to #include “.h” as we did in the previous lab

Like in the example declare an array in main(). Declare a MAXSIZE in main, then pass the parameters. Remember that you do not pass an array by reference, but if you are going to change the size variable when you Add, that will need to be passed by reference (& before the variable name)

When you add movies, you should add them to the next point in the array and increment the size, but should verify that you are not trying to add past the MAXSIZE of the array.

* Run the program and take a **screenshot** of the output. Place it below
* Turn in yourlastnamefirstLab07.cpp to canvas along with this lab that has 4 screen shots .