**Lab 3**

**Name: \_\_Jorge A. Serrano\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID#: \_121260\_\_\_\_\_\_\_\_\_\_\_\_**

1. Copy the source code developed for Lab 3 and paste it as **text** below. (*15 points*)

Header

#pragma once

/\*

\* CECS 2223, Computer Programming II Lab

\* Fall 2022, Sec. 05

\* Date: August 31, 2022

\* Topic: Lab 3 - Private Methods and Class Variables

\* File name: Cars.h

\* This file DECLARES a class named Cars

\*/

#include <iostream>

using namespace std;

class Cars {

private:

static int count; // The quantity of Cars objects created

static size\_t brandLength; // the length of the brand field

static size\_t modelLength; // the length of the model field

string brand; // the car's brand name

string model; // the car's model name

string serialNumber; // the car's serial number

string setSerialNumber(); // private method to create the serial number

void setBrandLength(); // private method to set the length of the brand field

void setModelLength(); // private method to set the length of the model field

public:

Cars(); // the default constructor

Cars(string, string); // parameterized constructor

void setBrand(string); // set the car's brand

void setModel(string); // set the car's model

string getBrandName() const; // gets the car's brand name

string getModelName() const; // gets the car's model name

string getSerialNumber() const; // gets the car's serial number

size\_t getBrandLength() const; // gets the brand field's length

size\_t getModelLength() const; // gets the model field's length

void printCar() const; // prints the car's information

};

Main

/\*

\* CECS 2223, Computer Programming II Lab

\* Fall 2022, Sec. 05

\* Date: August 31, 2022

\* Topic: Lab 3 - Private Methods and Class Variables

\* File name: FA-22\_lab03.cpp

\* This file IMPLEMENTS a class named Cars

\* Complete the code as required

\* YOUR NAME HERE, YOUR ID NUMBER HERE

\*/

// write the required include statement

#include "Cars.h"

string getBrand(); // method prototype

string getModel(); // method prototype

int main() {

// Declare 4 Cars objects, car1, car2, car3, and car4.

Cars car1, car2, car3, car4;

// call the getBrand and getModel method to assign values

// to all Cars objects

car1.setBrand(getBrand());

car1.setModel(getModel());

cout << endl;

car2.setBrand(getBrand());

car2.setModel(getModel());

cout << endl;

car3.setBrand(getBrand());

car3.setModel(getModel());

cout << endl;

car4.setBrand(getBrand());

car4.setModel(getModel());

cout << endl;

// Print the following table header:

// "BRAND", "MODEL", "SERIAL NUMBER"

// Remember to use the correct field size values

printf("%-10s%-10s%s\n", "BRAND", "MODEL", "SERIAL NUMBER");

// Call the print method for each Cars object to complete the table

car1.printCar();

car2.printCar();

car3.printCar();

car4.printCar();

cout << endl;

// write a statement which prints the phrase

// "Program developed by [YOUR NAME], ID#[YOUR ID NUMBER]"

// where the square brackets and the text within is substituted

// with your personal information.

cout << "Program developed by Jorge A. Serrano, ID#121260";

cout << endl;

system("pause"); // For Visual Studio use only!

return 0;

}

// METHOD DEFINITION AREA

// The getBrand method has no parameters and returns a string.

// It prompts the user for a brand name using the phrase

// "Enter the car's brand name: ", stores the value in a local

// variable named brand, and returns the stored value.

string getBrand()

{

string brand;

cout << "Enter the car's brand name: ";

cin >> brand;

return brand;

}

// The getModel method has no parameters and returns a string.

// It prompts the user for a model name using the phrase

// "Enter the car's model name: ", stores the value in a local

// variable named model,

string getModel()

{

string model;

cout << "Enter the car's model name: ";

cin >> model;

return model;

}

Cars.cpp

/\*

\* CECS 2223, Computer Programming II Lab

\* Fall 2022, Sec. 05

\* Date: August 31, 2022

\* Topic: Lab 3 - Private Methods and Class Variables

\* File name: Cars.cpp

\* This file DEFINES a class named Cars

\* Complete the code as required

\* YOUR NAME HERE, YOUR ID NUMBER HERE

\*/

// write the required include statement

#include "Cars.h"

// initialize the class variables to 0

int Cars::count = 0;

size\_t Cars::brandLength = 0;

size\_t Cars::modelLength = 0;

// Define the constructors. Each constructor adds 1 to

// the class variable. The parameterized constructor calls

// the setSerialNumber method.

Cars::Cars()

{

count++;

brand = "";

model = "";

serialNumber = "";

}

Cars::Cars(string b, string m)

{

count++;

brand = b;

model = m;

serialNumber = setSerialNumber();

}

// The setSerialNumber method creates the serial number for the car.

// The serial number is made up of the first three letters of the

// brand plus the first three letters of the model plus three numbers

// from the count. For example, for a Toyota Yaris with count 1 the

// serial number would be TOYYAR001

string Cars::setSerialNumber(){

string cereal = to\_string(count), brandName, modelName;

if (model.length() <= 0)

{

modelName = "";

}

else

{

modelName = model.substr(0, 3);

for (int i = 0; i < 3; i++)

{

modelName[i] = toupper(modelName[i]);

}

}

if (brand.length() <= 0)

{

brandName = "";

}

else

{

brandName = brand.substr(0, 3);

for (int i = 0; i < 3; i++)

{

brandName[i] = toupper(brandName[i]);

}

}

cereal.insert(cereal.begin(), 3 - cereal.length(), '0');

return (brandName + modelName + cereal);

}

// The setBrandLength and setModelLength methods determine the size of

// the corresponding field. The size of each field in the longest string

// plus 3 spaces.

void Cars::setBrandLength()

{

if (brand.size() + 3 > brandLength) {

brandLength = brand.size() + 3;

}

}

void Cars::setModelLength()

{

if (model.size() + 3 > modelLength) {

modelLength = model.size() + 3;

}

}

// Define the setters and getters.

// The setters for brand and model will call the field length methods.

void Cars::setBrand(string b)

{

brand = b;

setBrandLength();

serialNumber = setSerialNumber();

}

void Cars::setModel(string m)

{

model = m;

setModelLength();

serialNumber = setSerialNumber();

}

string Cars::getBrandName() const{

return brand;

}

string Cars::getModelName() const{

return model;

}

string Cars::getSerialNumber() const{

return serialNumber;

}

size\_t Cars::getBrandLength() const{

return brandLength;

}

size\_t Cars::getModelLength() const{

return modelLength;

}

// The printCar method prints the data for a car in a

// table ready format. The data to be printed is brand,

// model, and serial number. The information must be

// printed in a single line, make sure to add the line

// termination instruction.

// This method DOES NOT print the table header

void Cars::printCar() const {

if (getModelLength() != 0 && getBrandLength() != 0)

{

printf("%-\*s %-\*s % -s\n", getBrandLength(), getBrandName().c\_str(), getModelLength(), getModelName().c\_str(), getSerialNumber().c\_str());

}

else

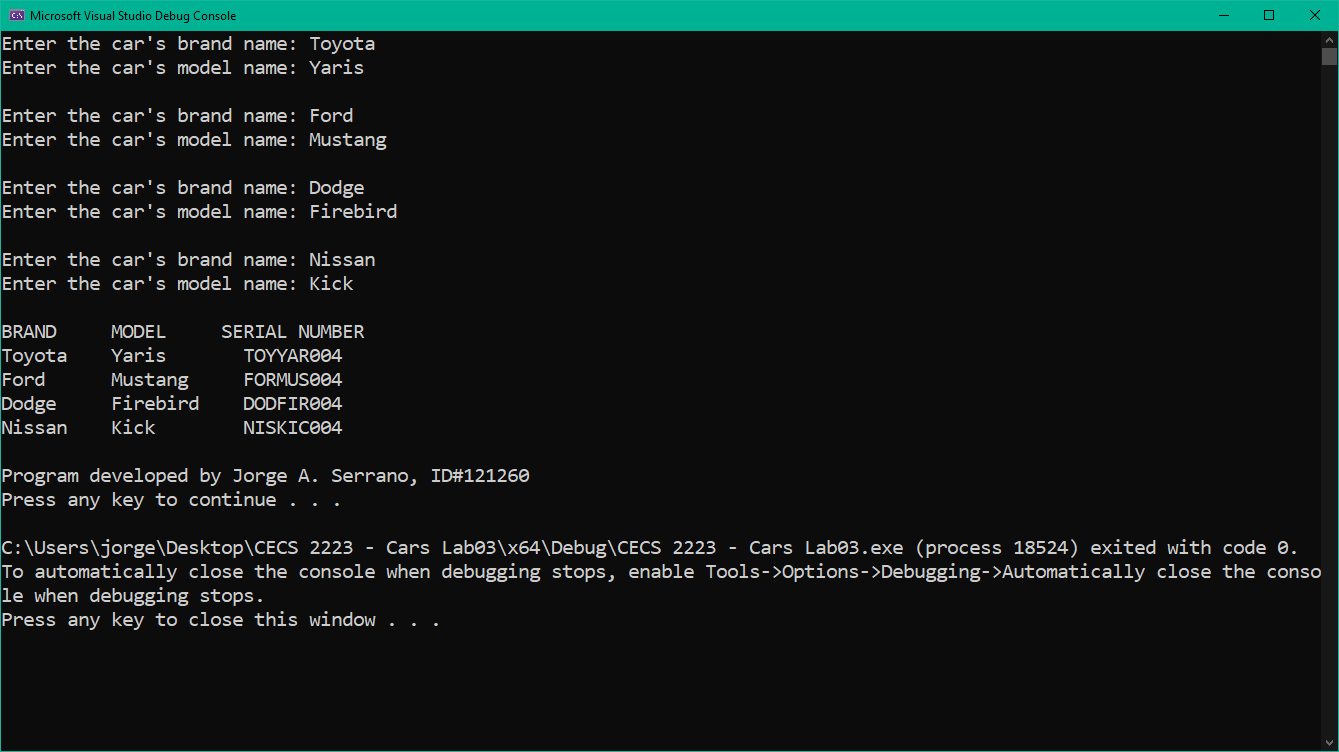
{

printf("DATA NOT AVAILABLE\n");

}

}

1. Paste the screenshots of the program’s execution below. (*5 points*)



1. Comment on any warnings or errors revealed by Visual Studio. If any error messages were present, list the error and describe how you corrected it. (*5 points*)

There were several problems but one that took me a while was the printf in which it didn’t print my serial number properly. When checked I didn’t include the .c\_str() at the end.