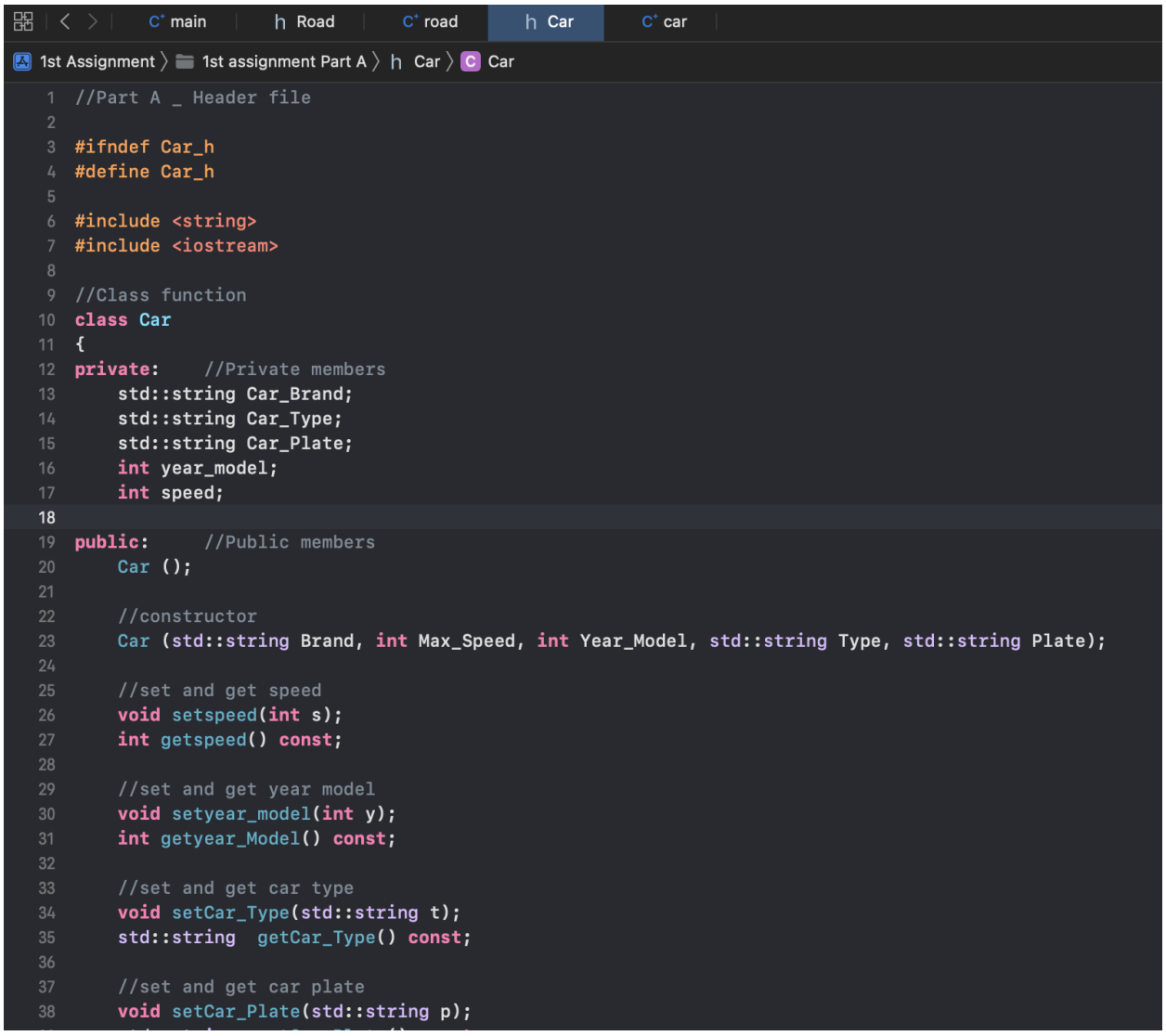
Report file for the first assignment

Abdelrahman Nashed
900211119
abdelrahman12@aucegypt.edu

Part A:

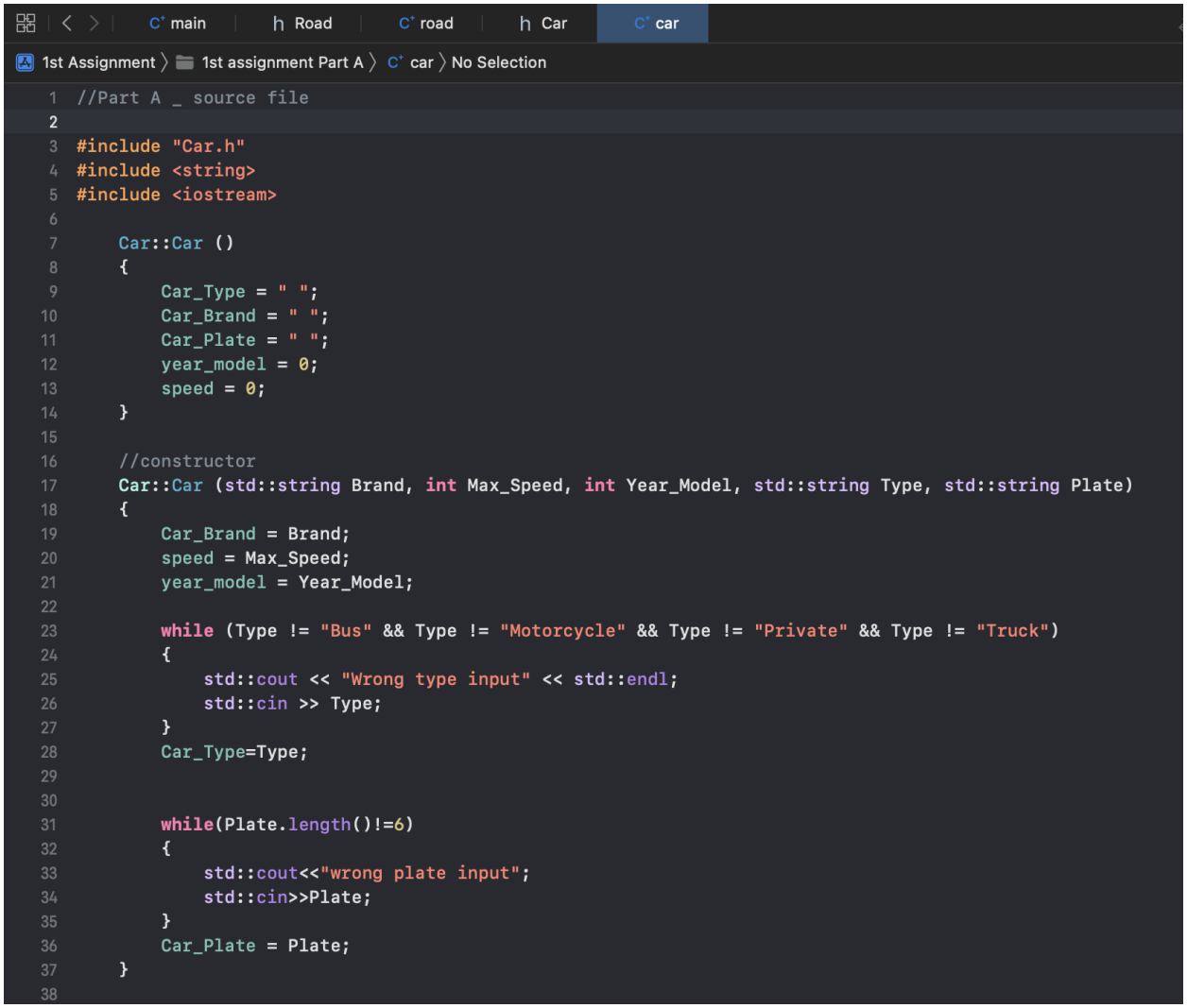
Car header file".h":




```
//Part A _ Header file

#ifndef Car_h
#define Car_h

#include <string>
#include <iostream>

//Class function
class Car
{
private:    //Private members
    std::string Car_Brand;
    std::string Car_Type;
    std::string Car_Plate;
    int year_model;
    int speed;

public:     //Public members
    Car ();

    //constructor
    Car (std::string Brand, int Max_Speed, int Year_Model, std::string Type, std::string Plate);

    //set and get speed
    void setspeed(int s);
    int getspeed() const;

    //set and get year model
    void setyear_model(int y);
    int getyear_Model() const;

    //set and get car type
    void setCar_Type(std::string t);
    std::string  getCar_Type() const;

    //set and get car plate
    void setCar_Plate(std::string p);
```

// I did not include using namespace std; for both Car and Road class to avoid declaring variables that share the same name. I read something like that before related to header. It says that it is better not to use using namespace std in the header files, but I do not know if it is the same for the cpp files or not. So, I just used std:: in both header and cpp files.

Car source file".cpp":



```cpp
//Part A _ source file

#include "Car.h"
#include <string>
#include <iostream>

    Car::Car ()
    {
        Car_Type = " ";
        Car_Brand = " ";
        Car_Plate = " ";
        year_model = 0;
        speed = 0;
    }

    //constructor
    Car::Car (std::string Brand, int Max_Speed, int Year_Model, std::string Type, std::string Plate)
    {
        Car_Brand = Brand;
        speed = Max_Speed;
        year_model = Year_Model;

        while (Type != "Bus" && Type != "Motorcycle" && Type != "Private" && Type != "Truck")
        {
            std::cout << "Wrong type input" << std::endl;
            std::cin >> Type;
        }
        Car_Type=Type;


        while(Plate.length()!=6)
        {
            std::cout<<"wrong plate input";
            std::cin>>Plate;
        }
        Car_Plate = Plate;
    }

```

// I created a cpp source file for Car class to be more organized

Part B:

Road header file".h":

/* There is no problem or any comment in the header file of Road except that I did not include using namespace std; and used std:: instead.*/

Road source file".cpp":

```
//allow function test to see which road (A,B, or C)
void Road::allow(std::string Car_Type)
{
    //Road A: Allow only Private and motorcycle vehicles.
    //Road B: Allow all vehicles
    //Road C: Allow only trucks.

        if ((Road_Type == 'A' || Road_Type == 'a')&&(Car_Type == "private" || Car_Type == "motorcycle"))
        {
            std::cout << "Private and motorcycle vehicles are allowed" << std::endl;
            countA++;
        }
        else if (Road_Type == 'B' || Road_Type == 'b')
        {
            std::cout << "All vehicles are allowed" << std::endl;
            countB++;
        }
        else if ((Road_Type == 'C' || Road_Type == 'c')&&(Car_Type == "truck"))
        {
            std::cout << "Truck vehicles are only allowed" << std::endl;
            countC++;
        }

}
```

/* In the second if-else statement I did not add any Car Type since all the vehicles are allowed in Road B.*/

// I also created a cpp source file for Road class to be more organized

Part C:
Main cpp file:

```
    Car cars [20];

    for (x = 0; x < 8; x++)
    {
        q.push(cars[x]);
    }
    for (y = 0; y < 10; y++)
    {
        q.push(cars[y]);
    }
    for (z = 0; z < 2; z++)
    {
        q.push(cars[z]);
    }
```

/* I created an array that takes 20 cars and added a number of cars randomly for each road to check at the end which one has the highest efficiency.*/

```
    efficiency1 = (x / 20.0) * 100.0;
    efficiency2 = (y / 20.0) * 100.0;
    efficiency3 = (z / 20.0) * 100.0;

    x = efficiency1;
    y = efficiency2;
    z = efficiency3;

    cout << "The efficiency of each road:" << endl;
    cout << "Road A: " << x << "%" << endl;
    cout << "Road B: " << y << "%" << endl;
    cout << "Road C: " << z << "%" << endl << endl;

    //Use if statement to determine the heighest efficiency
    if (x > y && x > z)
        max = x;
    else if (y > x && y > z)
        max = y;
    else if ( z > y && z > x)
        max = z;

    cout << "The road with the highest efficiency is Road: " << "B with " << max << " % efficiency."<< endl <<
        endl;
```

/* First of all, I calculated the efficiency of each road then created an if-else statement to see which road has the highest efficiency by declaring a variable called max.*/

Output:

```
This car will be fined
This car will be fined
This car will be fined
Private and motorcycle vehicles are allowed
Private and motorcycle vehicles are allowed
All vehicles are allowed
All vehicles are allowed
Truck vehicles are only allowed

The speed of the 1st Car: 80
The year model of the 1st Car: 2018
The speed of the 2nd Car: 80
The year model of the 2nd Car: 2019
The speed of the 3rd Car: 100
The year model of the 3rd Car: 2021
The speed of the 4th Car: 90
The year model of the 4th Car: 2020

In Road A, there are: 8 cars
In Road B, there are: 10 cars
In Road C, there are: 2 cars

The efficiency of each road:
Road A: 40%
Road B: 50%
Road C: 10%

The road with the highest efficiency is Road: B with 50 % efficiency.

Program ended with exit code: 0
```

/*The output shows 3 cars that exceeded the speed limit
Some cars that are allowed to pass
Some cars' contents such as their speed and year model
It shows how many car entered each road
Then I took these cars to calculate the efficiency out of 20
The road with the highest efficiency will be Road be since it contains 50% of the cars.*/