

# Sabanci University

Faculty of Engineering and Natural Sciences  
CS204 Advanced Programming  
Fall 2021

Homework 5 – Car sharing  
Due: 10/12/2021 (Friday), 09:00 am

## PLEASE NOTE:

**Your program should be a robust one such that you have to consider all relevant user mistakes and extreme cases; you are expected to take actions accordingly!**

**You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!**

## Introduction

This homework aims to practice on **object sharing via reference variables**. In this homework, you are asked to develop an IT system for a taxi company where two or more drivers share a taxi vehicle (car). In your implementation you are going to write two classes, Car and Driver, such that two (or more) driver objects share one car object via reference variables. The main function of the program is already given to you. We will explain details about the homework in the following sections.

## Using Object Sharing Principles and Object Oriented Design

As mentioned above, you have to have one Car and two Driver objects in your program. The Car object must be shared by the Driver objects. For this object sharing, you have to employ the method that uses reference variables.

**From the above paragraph it should be clear that you will write two classes for Car and Driver. You need to analyze the requirements carefully and make a good object oriented design for these classes. In this context, you have to determine the data members and member functions of each class correctly. We will evaluate your object oriented design as well. Moreover, you are not allowed to use friend class or friend functions in your design.**

## Program Flow

**At the beginning of the main function, which is provided with the homework package, initial values of the Car object sharedCar are set to 300\$ for fuel level (which is the maximum fuel level that the car can have), 1000\$ for insurance fee and 5000km for the total distance that the car has travelled. Initial budgets of Driver objects, driver1 and driver2, are set to 1000\$.**

The specifications of member functions of Drive class are as follows:

1. `void drive (int)`: The function takes the kilometer(km) that the driver has driven so far as an input. Assume that the km is an integer value. The function should increase the total distance that the car has travelled by this km. Also, the function should reduce the fuel level of the car. You can assume that the car consumes fuel corresponds to 0.25\$ for each km.
2. `void repairCar(string)`: Each driver may have an accident with the car. There are 3 types of accidents: SMALL, MEDIUM and LARGE depending on the damage that the car had. The `repairCar(string)` function is called by one of the drivers, the budget of the driver should reduce 50\$ for SMALL accidents, 150\$ for MEDIUM accidents, and 300\$ for LARGE accidents. In each accident, the insurance fee of the car should increase by 5%, 10% and 20% for SMALL, MEDIUM, and LARGE accidents, respectively.
3. `void display()`: The function should display the current budget of driver.
4. `void fullFuel()`: The function should full the fuel of the car to its maximum level (300\$). Please note that when a driver fills up the fuel, the budget of the driver should decrease accordingly.

The specifications of member functions of Car class are as follows:

1. `void display()`: The function should display the fuel level, insurance fee and the total distance that the car has travelled so far as shown in the sample runs.

The only member function of Car class that is called from the main function is `display()`. There may be other member functions of Car class that are called by a member function of Driver class. You should be aware of such functions and write them appropriately.

Please note that you can assume all inputs are entered correctly, so you do not need any input check in your program. You can also assume that the budget of a driver and fuel level of the car reduces below zero. Therefore, you do not have to check such cases in your implementation as well.

## Sample Runs

**Sample Run 1 (user inputs are given in bold letters):**

How many km driver1 drives?  
**100**

Did the driver1 have an accident YES / NO  
**YES**

What is the type of the accident SMALL / MEDIUM / LARGE  
**MEDIUM**

150\$ is reduced from the driver's budget because of the MEDIUM accident  
Yearly insurance fee is increased to 1100 because of the MEDIUM accident

Fuel Level: 275  
Insurance Fee: 1100  
Total distance that the car has travelled: 5100

Fuel is full

Driver Budget: 825

How many km driver2 drives?  
**200**

Did the driver2 have an accident YES / NO  
**NO**

Fuel Level: 250  
Insurance Fee: 1100  
Total distance that the car has travelled: 5300

Fuel is full

Driver Budget: 950

Do you want to continue: YES / NO  
**YES**

How many km driver1 drives?  
**250**

Did the driver1 have an accident YES / NO  
**NO**

Fuel Level: 237.5

Insurance Fee: 1100  
Total distance that the car has travelled: 5550  
Fuel is full

Driver Budget: 762.5

How many km driver2 drives?  
**50**

Did the driver2 have an accident YES / NO  
**YES**

What is the type of the accident SMALL / MEDIUM / LARGE  
**LARGE**

300\$ is reduced from the driver's budget because of the LARGE accident  
Yearly insurance fee is increased to 1320 because of the LARGE accident

Fuel Level: 287.5  
Insurance Fee: 1320

Total distance that the car has travelled: 5600  
Fuel is full

Driver Budget: 637.5

Do you want to continue: YES / NO  
**NO**

Press any key to continue . . .

### **Sample Run 2 (user inputs are given in bold letters):**

How many km driver1 drives?  
**40**

Did the driver1 have an accident YES / NO  
**YES**

What is the type of the accident SMALL / MEDIUM / LARGE  
**LARGE**

300\$ is reduced from the driver's budget because of the LARGE accident  
Yearly insurance fee is increased to 1200 because of the LARGE accident

Fuel Level: 290  
Insurance Fee: 1200  
Total distance that the car has travelled: 5040  
  
Fuel is full

Driver Budget: 690

How many km driver2 drives?  
**500**

Did the driver2 have an accident YES / NO  
**YES**

What is the type of the accident SMALL / MEDIUM / LARGE  
**SMALL**

50\$ is reduced from the driver's budget because of the SMALL accident  
Yearly insurance fee is increased to 1260 because of the SMALL accident

Fuel Level: 175  
Insurance Fee: 1260  
Total distance that the car has travelled: 5540  
  
Fuel is full

Driver Budget: 825

Do you want to continue: YES / NO  
**YES**

How many km driver1 drives?  
**140**

Did the driver1 have an accident YES / NO  
**NO**

Fuel Level: 265  
Insurance Fee: 1260  
Total distance that the car has travelled: 5680

Fuel is full

Driver Budget: 655

How many km driver2 drives?  
**80**

Did the driver2 have an accident YES / NO  
**NO**

Fuel Level: 280  
Insurance Fee: 1260  
Total distance that the car has travelled: 5760

Fuel is full

Driver Budget: 805

Do you want to continue: YES / NO  
**NO**

Press any key to continue . . .

### **Some Important Rules**

Although some of the information is given below, first, please read the homework submission and grading policies in the course webpage and lecture notes of the first week. In order to get a full credit, your programs must be efficient and well commented and indented. Presence of any redundant computation or bad indentation, or missing, irrelevant comments may decrease your grades if we detect them. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homework we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in *Release* mode and **we may test your programs with very large test cases**.

### **What and where to submit (PLEASE READ, IMPORTANT)**

You should prepare (or at least test) your program using MS Visual Studio 2012 C++. We will use the standard C++ compiler and libraries of the abovementioned platform while testing your homework. You need to place your first and last name in the program (as a comment line of course).

Submissions guidelines are below. Some parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Name your cpp file that contains your program as follows:

*"SUCourseUserName\_YourLastname\_YourName\_HWnumber.cpp"*

Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsizkodyazaroglu, then the file name must be :

*Cago\_Ozbugsizkodyazaroglu\_Caglayan\_hw5.cpp*

Do not add any other character or phrase to the file name. Make sure that this file is the latest version of your homework program. Compress this cpp file using WINZIP or WINRAR programs. Please use "zip" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension. Check that your compressed file opens up correctly and it contains your cpp file.

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

You will receive no credits if your compressed zip file does not expand or it does not contain the correct file. The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows:

*SUCourseUserName\_YourLastname\_YourName\_HWnumber.zip*

For example, zubzipler\_Zipleroglu\_Zubeyir\_hw5.zip is a valid name, but

*Hw5\_hoz\_HasanOz.zip, HasanOzHoz.zip*

are **NOT** valid names.

**Submit via SUCourse ONLY!** You will receive no credits if you submit by other means (email, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

**Note:** You shouldn't deal with extra input checks. For consistency, the questions related to HW5 should be asked to your TA Seyedpouya Seyedkazemi or the lecturer. Of course, technical help regarding HW5 can be obtained from the TAs during their OH and the lecturer. Your submitted zip file should include all of the headers .h and the corresponding .cpp files, if you will have them, of course.

Good Luck!

CS204 Team (Artrim Kjamilji, Seyedpouya Seyedkazemi)