CAR Rental Simulator

Project Description:

- It is required to develop a program that simulates the Car Rental platform.
- The code should be organized in 3 files: **vehicle.h**, **customer.h**, **main.cpp**
- The **vehicle.h file** will contain class definitions (data members and member function).
- The **customer.h** file will contain class definitions (data members and member function).
- The **main.cpp** file should contain objects creation and other needed functions.

Operations:

(1) Vehicle.h and vehicle.cpp files details:

Operation	**Sub Operation**	**Action Required**
class CVehicle	Private variables	int Car_Number, char car_model[10], char car_type[10], float car_price[5], bool rented, char return_time[10], char rentled_name[50]…. Ect.
	Public variables	**NONE**. Create **setter** and **getter** functions to access all private variables
	Constructor	Initialize each variable with NULL for string and Zero for other numbers.
	Add_Car_details()	A public function that 1) prompts the user to enter car data, and 2) stores these values for the Cvehicle class members.
	Get_cars_Info()	A public function that prints all vehicle ‘s data.
	Rent_car()	Make the boolen value True and write the return time
Class **CCustomer**	Private variables	int Cus_ID, char Cus_name[50], char Cus_Email[10], char Email_Password[10]…. Ect.
	Public variables	**NONE**. Create **setter** and **getter** functions to access all private variables
	Constructor	

	Add_customer_info()
	Get_Customer_info()
		A public function that accepts values for the customer class members and store them.
		A public function that prints the customer details.

class CBUS (BONUS) (A class for Rent Bus. This class inherits from vehicle class and adds a new data member: number_passengers _of_bus , name_of_driver)	private variables public variables	This class should **inherit** the vehicle class and adds another private data member: number_passengers _of_bus , name_of_driver. **NONE**. Create **setter** and **getter** functions to access all private variables for the CBUS class.
(2) Main.cpp file:		
		1- First, input the number of cars (n_cars) to be add in the system 2- Then create an array of Cvehicle objects, anther array of Ccustomer 3- Create a for loop from 1 to n_cars that calls Add_Car_details() function to store Cars' data. 4- Create another for loop from 1 to n_cars that calls Get_cars_Info(). 5- Make a user to login () and then rent a car, 6- Then print all the cars details in user friendly way in console. 7- (**Bonus**) Inherit the class Bus from the vehicle class and perform the same operations for the inherited class as in 1,2,3,4 while taking into consideration the new data member (no. of passenger, and the driver name).

General Constraints:

- All your code should be **in one folder**.
- Do not clear the screen after every operation.
- Just **one of the team members must submit the file**, with a comment inside the main function with student names.

Grading Rubrics:

- Specifications: The program works and meets all the requirements (55%).
- Readability: The code is well organized and easy to follow (15%).
- Documentation: The code is well documented and clearly explained (20%).
- Delivery: The program was delivered on time (10%).
- Bonus: 20%

Submission:

Each project submission (on Moodle) must include:

1- A whole code project's folder (zipped).

2- A report with the following:

a. Team member names and IDs

b. Application description

c. Flowchart of execution sequence

d. Sample input and output screens.

Due Date:

30th JUN 2022, 11:00pm

Late submissions will be penalized.

Discussion Date:

2nd JUL 2022.

Teams:

Work in groups of 4-5 students.

Plagiarism:

Plagiarism is a serious academic offence and students who share code with others or get any source from the internet will fail the course. A plagiarism detection tool will be used to check all projects submitted and check and report plagiarism case.