**Online Car Rental Platform.**

**Description**

Build an online car rental platform using Object-Oriented Programming in Python.

**Problem Statement:**

A car rental company has requested you to build an online car rental platform where customers should be able to view the available cars that can be rented on an hourly, daily, or weekly basis. The company can display the available inventory and confirm requests by checking the available stock. Customers will receive an auto-generated bill when they return the car.

For simplicity, let’s assume that:

1. Customers can rent cars from any one of the following options—hourly, daily, or weekly rental.
2. Customers are free to choose any number of cars they want, provided the number of available cars is more than the number of requested cars.

**Instructions to Perform:**

1. Create a module (.py file) for car rental and import the built-in module DateTime to handle the rental time and bill.
2. Create a class for renting the cars and define a constructor in it.
3. Define a method for displaying the available cars. Also, define methods for renting cars on an hourly, daily and weekly basis, respectively.
4. Inside these methods, make sure that the number of requested cars is positive and lesser than the total available cars.
5. Store the time of renting a car in a variable, which can later be used in the bill while returning the car.
6. Define a method to return the cars using rental time, rental mode (hourly, daily, or weekly), and the number of cars rented.
7. Inside the return method; update the inventory stock, calculate the rental period, and generate the final bill.
8. Create a class for customers and define a constructor in it.
9. Define methods for requesting the cars and returning them.
10. Next, create the main project (.ipynb) file and import the car rental module.
11. Define the main method and create objects for both car rental and customer classes.
12. Inside the main method, take the customer’s input as a choice for displaying car availability, rental modes, or returning the cars.
13. Use the relevant method for the customer’s input and print relevant messages.
14. Run the main method to start your project.

**import datetime as dt**

here the datetime module is imported as dt

**Bottom Code is the Class for rental of car based on rental duration.**

**class CarRental:**

**def \_\_init\_\_(self):**

**self.available\_cars =["SUV","Sedan","Hatchback"]**

**def display\_car\_availability(self):**

**print("Available cars for rent:")**

**for car in self.available\_cars:**

**print(car)**

**def rent\_hourly(self, hours, requested\_cars):**

**if requested\_cars > 0 and requested\_cars <= self.available\_cars:**

**print(f"You have rented {requested\_cars} car(s) for {hours} hours.")**

**else:**

**print("Invalid number of requested cars. Please try again.")**

**def rent\_daily(self, days, requested\_cars):**

**if requested\_cars > 0 and requested\_cars <= self.available\_cars:**

**print(f"You have rented {requested\_cars} car(s) for {days} days.")**

**else:**

**print("Invalid number of requested cars. Please try again.")**

**def rent\_weekly(self, weeks, requested\_cars):**

**if requested\_cars > 0 and requested\_cars <= self.available\_cars:**

**print(f"You have rented {requested\_cars} car(s) for {weeks} weeks.")**

**else:**

**print("Invalid number of requested cars. Please try again.")**

**def calculate\_bill(rental\_time, rental\_mode, num\_cars):**

**if rental\_mode == "hourly":**

**rate = 10**

**total\_cost = rental\_time \* rate \* num\_cars**

**elif rental\_mode == "daily":**

**rate = 50**

**total\_cost = rental\_time \* rate \* num\_cars**

**elif rental\_mode == "weekly":**

**rate = 200**

**total\_cost = rental\_time \* rate \* num\_cars**

**else:**

**return "Invalid rental mode!"**

**return total\_cost**

**def return\_car(inventory, rental\_start\_time, rental\_end\_time, rental\_rate):**

**# Update inventory stock**

**inventory -= 1**

**# Calculate rental period**

**rental\_period = rental\_end\_time - rental\_start\_time**

**# Generate the final bill**

**final\_bill = rental\_period \* rental\_rate**

**return inventory, rental\_period, final\_bill**

**Bottom Code is the making of class customers along with request car and return car methods in it.**

**class Customer:**

**def \_\_init\_\_(self, name):**

**self.name = name**

**self.rented\_cars = []**

**def request\_car(self, car):**

**self.rented\_cars.append(car)**

**print(f"{self.name} has requested the {car}.")**

**def return\_car(self, car):**

**if car in self.rented\_cars:**

**self.rented\_cars.remove(car)**

**print(f"{self.name} has returned the {car}.")**

**else:**

**print(f"{self.name} did not rent the {car}.")**

After the execution of both the classes along with datetime function, we now download and export the cars file as executable script.

After downloading of the cars file, there is creation of the project module and we import our cars.py(executable script of cars module) in it.

**import cars**

This is the importing of our cars file in our project module.

**car\_rental=cars.CarRental()**

car\_rental is the object of CarRental class

**customer = cars.Customer("Bharat")**

customer is the object of Customer class

**def main():**

**while True:**

**print("What would you like to do?")**

**print("1. Display car availability")**

**print("2. Rent a car")**

**print("3. Return a car")**

**print("4. Exit")**

**choice = input("Enter your choice (1-4): ")**

**if choice == "1":**

**car\_rental.display\_car\_availability()**

**elif choice == "2":**

**car\_choice = input(f"{customer.name}, enter the car you want to rent: ")**

**customer.request\_car(car\_choice)**

**elif choice == "3":**

**car\_choice = input(f"{customer.name}, enter the car you want to return: ")**

**customer.return\_car(car\_choice)**

**elif choice == "4":**

**print(f"Thank you, {customer.name}, for using our car rental service!")**

**break**

**else:**

**print("Invalid choice. Please try again.").**

The above method is the main method that will ask about the choice of customer’s input as a choice for displaying car availability, rental modes, or returning the cars.

**main()**

The above main() method allows us to call our main method.

After calling our main method, the customer gets to choose from 4 options.



In this,

option 1 is the availability of cars.

option 2 is which car the customer wants to rent.

option 3 is which rented car the customer wants to return

option 4 is of exit where after using our services we greet the customer.

The bottom pdf files are the code of both the files that are used to complete this project.

