

Source code

car_rent_platform.ipynb 1st of two

```
Launcher x car_rent_platform.ipynb x car_rental.py x
+ ✂ 📄 ▶ ⏏ ⌂ ⏪ ⏩ Code v

[ ]: from car_rental import CarRental, Customer

shop = CarRental(1000)
customer = Customer()

while True:
    print('''
        Welcome to our shop!
        1. Display available cars.
        2. Request a car on hourly basis, the fee will charge $10 one hour/car.
        3. Request a car on daily basis, the fee will charge $180 one day/car.
        4. Request a car on weekly basis, the fee will charge $900 one week/car.
        5. Return a car (or cars).
        6. Exit.
        ''')

    choice = input("Enter choice: ")

    try:
        choice = int(choice)

    except ValueError:
        print("That's an invalid input. Please enter number between 1-6.")

    if choice == 1: # stock
        shop.display_stock()

    elif choice == 2: # hourly
        customer.rental_time = shop.rent_car_hourly(customer.request_car())
        customer.rental_basis = 1

    elif choice == 3: # daily
        customer.rental_time = shop.rent_car_hourly(customer.request_car())
        customer.rental_basis = 2

    elif choice == 4: # weekly
        customer.rental_time = shop.rent_car_hourly(customer.request_car())
        customer.rental_basis = 3

    elif choice == 5: # return a car
        customer.bill = shop.return_car(customer.return_car())
        customer.rental_basis, customer.rental_time, customer.cars = 0, 0, 0

    elif choice == 6: # Exit
        break
```

```

Launcher X car_rent_platform.ipynb X car_rental.py X
Code
elif choice == 4: # weekly
    customer.rental_time = shop.rent_car_hourly(customer.request_car())
    customer.rental_basis = 3

elif choice == 5: # return a car
    customer.bill = shop.return_car(customer.return_car())
    customer.rental_basis, customer.rental_time, customer.cars = 0, 0, 0

elif choice == 6: # Exit
    break

else:
    print("That's an invalid input. Please enter number between 1-6.")

print("Thank you for using our car rental system.")

```

-

```

Welcome to our shop!
1. Display available cars.
2. Request a car on hourly basis, the fee will charge $10 one hour/car.
3. Request a car on daily basis, the fee will charge $180 one day/car.
4. Request a car on weekly basis, the fee will charge $900 one week/car.
5. Return a car (or cars).
6. Exit.

Enter choice: 1
We currently have 1000 cars available to rent.

```

-

```

Welcome to our shop!
1. Display available cars.
2. Request a car on hourly basis, the fee will charge $10 one hour/car.
3. Request a car on daily basis, the fee will charge $180 one day/car.
4. Request a car on weekly basis, the fee will charge $900 one week/car.
5. Return a car (or cars).
6. Exit.

Enter choice: 2
How many cars would you like to rent?4
You have rented a 4 car(s) on hourly basis today at 0 hours.
You will be charged $10 for each hour per car.
We hope that you enjoy our service.

```

-

```

Welcome to our shop!
1. Display available cars.
2. Request a car on hourly basis, the fee will charge $10 one hour/car.
3. Request a car on daily basis, the fee will charge $180 one day/car.
4. Request a car on weekly basis, the fee will charge $900 one week/car.
5. Return a car (or cars).
6. Exit.

```

car_rental.py 1st of three

```

1 import datetime
2
3 class CarRental: # class to build car rental shop.
4
5     def __init__(self,stock=0):
6         self.stock = stock
7
8     def display_stock(self): # display available cars for rent currnetly.
9         print("We currently have {} cars available to rent.".format(self.stock))
10        return self.stock
11
12    def rent_car_hourly(self,n): # rent "n" cars hourly
13        if n <= 0:
14            print("Sorry, the number of cars should be positive.")
15            return None
16
17        elif n > self.stock:
18            print("Sorry, we currently have {} cars available to rent.".format(self.stock))
19            return None
20
21        else:
22            now = datetime.datetime.now()
23            print("You have rented a {} car(s) on hourly basis today at {} hours.".format(n,now.hour))
24            print("You will be charged $10 for each hour per car.")
25            print("We hope that you enjoy our service.")
26
27            self.stock -= n
28            return now
29
30    def rent_car_daily(self,n): # rent "n" cars daily
31        if n <= 0:
32            print("Sorry, the number of cars should be positive.")
33            return None
34
35        elif n > self.stock:
36            print("Sorry, we currently have {} cars available to rent.".format(self.stock))
37            return None
38
39        else:
40            now = datetime.datetime.now()
41            print("You have rented a {} car(s) on hourly basis today at {} hours.".format(n, now.hour))
42            print("You will be charged $180 for each day per car.")
43            print("We hope that you enjoy our service.")
44
45            self.stock -= n
46            return now
47

```

```
Launcher x car_rent_platform.ipynb x car_rental.py x
46         return now
47
48     def rent_car_weekly(self,n):    # rent "n" cars weekly
49         if n <= 0:
50             print("Sorry, the number of cars should be positive.")
51             return None
52
53         elif n > self.stock:
54             print("Sorry, we currently have {} cars available to rent.".format(self.stock))
55             return None
56
57         else:
58             now = datetime.datetime.now()
59             print("You have rented a {} car(s) on hourly basis today at {} hours.".format(n, now.hour))
60             print("You will be charged $900 for each week per car.")
61             print("We hope that you enjoy our service.")
62
63             self.stock -= n
64             return now
65
66     def return_car(self,request):    # return cars from customer, rebalance the stock, make a bill
67         rental_time, rental_basis, num_of_cars = request
68         bill = 0
69         if rental_time and rental_basis and num_of_cars:
70             self.stock += num_of_cars
71             now = now = datetime.datetime.now()
72             rental_period = now - rental_time
73
74             # hourly bill calculation
75             if rental_basis == 1:
76                 bill = round(rental_period.seconds / 3600) * 10 * num_of_cars
77
78             # daily bill calculation
79             elif rental_basis == 2:
80                 bill = round(rental_period.days) * 180 * num_of_cars
81
82             # weekly bill calculation
83             elif rental_basis == 3:
84                 bill = round(rental_period.days / 7) * 900 * num_of_cars
85
86             print("Thanks for returning your car.")
87             print("That would be ${}".format(bill))
88             print("Hope you enjoy our service.")
89             return bill
90         else:
91             print("Are you sure you rented a car with us?")
92             return None
93
```

```
Launcher x car_rent_platform.ipynb x car_rental.py x
92         return None
93
94
95 class Customer: # class to build customer object.
96
97     def __init__(self): #
98         self.cars = 0
99         self.rental_basis = 0
100        self.rental_time = 0
101        self.bill = 0
102
103    def request_car(self): # num of cars request
104        cars = input("How many cars would you like to rent?")
105        try:
106            cars = int(cars)
107        except ValueError:
108            print("Sorry, that's not a positive integer!")
109            return -1
110
111        if cars < 1:
112            print("Sorry, the number of cars should be positive.")
113            return -1
114        else:
115            self.cars = cars
116        return self.cars
117
118    def return_car(self): # return cars
119        if self.rental_basis and self.rental_time and self.cars:
120            return self.rental_time, self.rental_basis, self.cars
121        else:
122            return 0, 0, 0
```