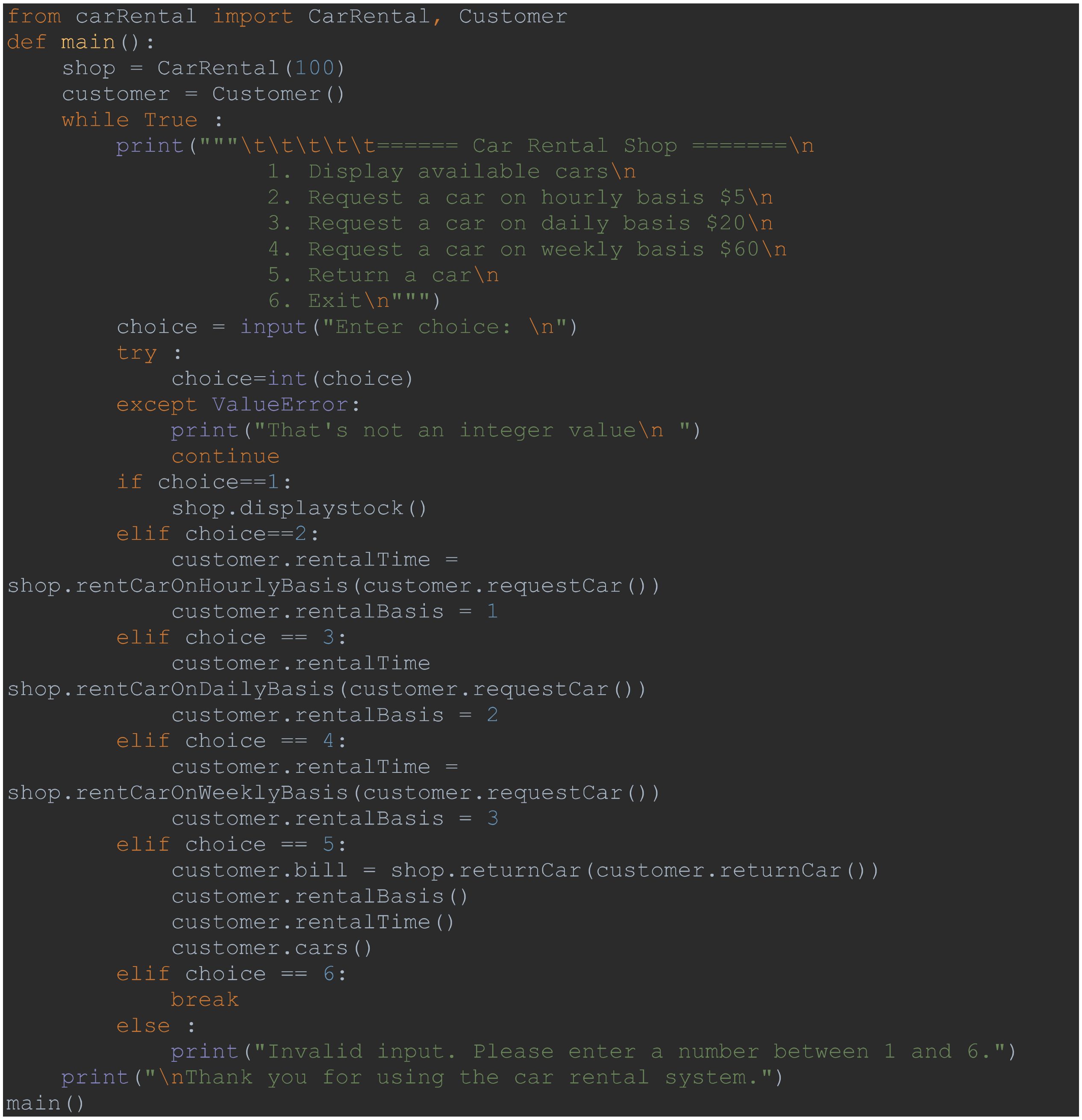

```
from carRental import CarRental, Customer
def main():
    shop = CarRental(100)
    customer = Customer()
    while True :
        print("""\t\t\t\t\t====== Car Rental Shop =======\n
                   1. Display available cars\n
                   2. Request a car on hourly basis $5\n
                   3. Request a car on daily basis $20\n
                   4. Request a car on weekly basis $60\n
                   5. Return a car\n
                   6. Exit\n""")
        choice = input("Enter choice: \n")
        try :
            choice=int(choice)
        except ValueError:
            print("That's not an integer value\n ")
            continue
        if choice==1:
            shop.displaystock()
        elif choice==2:
            customer.rentalTime =
shop.rentCarOnHourlyBasis(customer.requestCar())
            customer.rentalBasis = 1
        elif choice == 3:
            customer.rentalTime
shop.rentCarOnDailyBasis(customer.requestCar())
            customer.rentalBasis = 2
        elif choice == 4:
            customer.rentalTime =
shop.rentCarOnWeeklyBasis(customer.requestCar())
            customer.rentalBasis = 3
        elif choice == 5:
            customer.bill = shop.returnCar(customer.returnCar())
            customer.rentalBasis()
            customer.rentalTime()
            customer.cars()
        elif choice == 6:
            break
        else :
            print("Invalid input. Please enter a number between 1 and 6.")
    print("\nThank you for using the car rental system.")
main()
```

```
import datetime


class CarRental:

    def __init__(self, stock=0):
        """
        Constructor class to instantiate car rental shop.
        """

        self.stock = stock

    def displaystock(self):
        """
        Displays the currently available cars for rent.
        """

        print("We have currently {} cars available to
rent.".format(self.stock))
        return self.stock

    def rentCarOnHourlyBasis(self, n):
        """
        Rent a car on hourly basis to a customer.
        """
        if n <= 0:
            print("Number of cars should be positive!")
            return None

        elif n > self.stock:
            print("Sorry! We have currently {} cars available to
rent.".format(self.stock))
            return None

        else:
            now = datetime.datetime.now()
            print("You have rented a {} car(s) on hourly basis today at {}
hours.".format(n, now.hour))
            print("You will be charged $5 for each hour per car.")
            print("We hope that you enjoy our service.")

            self.stock -= n
            return now

    def rentCarOnDailyBasis(self, n):
        """
        Rents a car on daily basis to a customer.
        """
        if n <= 0:
            print("Number of cars should be positive!")
            return None

        elif n > self.stock:
            print("Sorry! We have currently {} cars available to
rent.".format(self.stock))
            return None
        else:
            now = datetime.datetime.now()
            print("You have rented {} car(s) on daily basis today at {}
hours.".format(n, now.hour))
            print("You will be charged $20 for each day per car.")
            print("We hope that you enjoy our service.")

            self.stock -= n
            return now

    def rentCarOnWeeklyBasis(self, n):
        """
        Rents a car on weekly basis to a customer.
        """
        if n <= 0:
            print("Number of cars should be positive!")
            return None

        elif n > self.stock:
            print("Sorry! We have currently {} cars available to
rent.".format(self.stock))
            return None

        else:
            now = datetime.datetime.now()
            print("You have rented {} car(s) on weekly basis today at {}
hours.".format(n, now.hour))
            print("You will be charged $60 for each week per car.")
            print("We hope that you enjoy our service.")
            self.stock -= n

            return now

    def returnCar(self, request):
        """
        1. Accept a rented car from a customer
        2. Replensihes the inventory
        3. Return a bill
        """
        rentalTime, rentalBasis, numOfCars = request
        bill = 0

        if rentalTime and rentalBasis and numOfCars:
            self.stock += numOfCars
            now = datetime.datetime.now()
            rentalPeriod = now - rentalTime

            # hourly bill calculation
            if rentalBasis == 1:
                bill = round(rentalPeriod.seconds / 3600) * 5 * numOfCars

            # daily bill calculation
            elif rentalBasis == 2:
                bill = round(rentalPeriod.days) * 20 * numOfCars

            # weekly bill calculation
            elif rentalBasis == 3:
                bill = round(rentalPeriod.days / 7) * 60 * numOfCars

            if (3 <= numOfCars <= 5):
                print("You are eligible for Family rental promotion of 30%
discount")
                bill = bill * 0.7

            print("Thanks for returning your car. Hope you enjoyed our
service!")
            print("That would be ${}".format(bill))
            return bill
        else:
            print("Are you sure you rented a car with us?")
            return None


class Customer:

    def __init__(self):
        """
        Constructor method to instantiate various customer objects.
        """

        self.cars = 0
        self.rentalBasis = 0
        self.rentalTime = 0
        self.bill = 0

    def requestCar(self):
        """
        Takes a request from the customer for the number of cars.
        """

        cars = input("How many cars would you like to rent?")
        try:
            cars = int(cars)
        except ValueError:
            print("That's not a positive integer!")
            return -1

        if cars < 1:
            print("Invalid input. Number of cars should be greater than
zero!")
            return -1
        else:
            self.cars = cars
        return self.cars

    def returnCar(self):
        """
        Allows customers to return their cars to the rental shop.
        """
        if self.rentalBasis and self.rentalTime and self.cars:
            return self.rentalTime, self.rentalBasis, self.cars
        else:
            return 0, 0, 0
```