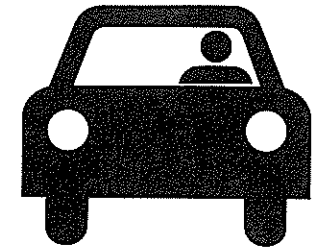


## COMPUTATIONAL PROBLEM SOLVING

### 10.6 Vehicle Rental Agency Program

In this section, we design, implement, and test a program that will serve the needs of a vehicle rental agency.



#### 10.6.1 The Problem

The problem is to develop an object-oriented design and implementation of a program capable of maintaining reservations for a vehicle rental agency. The agency rents out three types of vehicles—cars, vans, and moving trucks. The program should allow users to check for available vehicles, request rental charges by vehicle type, get the cost of renting a particular type vehicle for a specified period of time, and make/cancel reservations.

#### 10.6.2 Problem Analysis

The program needs an appropriate set of objects for the vehicle rental agency domain. An obvious class to include is a `Vehicle` class. It can be implemented to maintain information common to all vehicle types. Subclasses of the `Vehicle` class can maintain information specific to each subtype.

For example, all vehicles have a miles-per-gallon rating and a vehicle identification number (VIN). Thus, this information can be maintained in the `Vehicle` class. However, there are different make and model cars (with either two or four doors, that hold a specific number of passengers); different make and model vans (able to hold a specific number of passengers); and moving trucks of various lengths, each providing a certain amount of cargo space. Therefore, the `Vehicle` class is made a superclass of classes `Car`, `Van`, and `Truck`, in which each subclass contains information (instance variables and/or methods) specific to that vehicle type.

For each type vehicle, there is a rental charge based on daily, weekly, and weekend rental rates. There is also a mileage charge and some number of free miles (on select vehicles), plus the cost of optional insurance. Because these costs are associated with particular types, but cost is not inherently *part* of a vehicle's attribute, we include a separate `VehicleCost` class.

Finally, we incorporate a `Reservation` class that maintains the information for each reservation made. This will include the customer name, address, credit card number, and the VIN of the vehicle rented.

#### 10.6.3 Program Design

##### Meeting the Program Requirements

The general requirements for this program are for users to be able to check for the availability of vehicles of a certain type (cars, vans, or trucks); request rental charges by vehicle type; determine the rental cost for a particular vehicle and rental period; and make and cancel reservations. The specific requirements of this program are given in Figure 10-31.

The program must maintain a group of specific model vehicles for the following vehicle categories: cars, vans, and (moving) trucks with the following characteristics:

- Cars: make/model, miles-per-gallon, num of passengers, num of doors, VIN
- Vans: make/model, miles-per-gallon, number of passengers, VIN
- Trucks: miles-per-gallon, length, number of rooms, VIN

The program must be able to display the specific vehicles available for rent by vehicle type.

The program must display the cost associated with a given type vehicle including daily, weekend and weekly rate, insurance cost, mileage charge, and number of free miles. It must also allow the user to determine the cost of a particular vehicle, for a given period of time, an estimated number of miles, and the cost of optional insurance.

The program must be able to allow a particular vehicle to be reserved and cancelled.

FIGURE 10-31 Program Requirements for the Vehicle Rental Agency Program

The specific rental costs for each vehicle type are given in Figure 10-32.

	Daily Rate	Weekly Rate	Weekend Rate	Free Miles Per Day	Per Mile Charge
Car	\$24.99	\$180.00	\$45.00	100	0.15
Van	\$35.00	\$220.00	\$55.00	0	0.20
Truck	\$34.95	\$425.00	\$110.00	25	0.25

FIGURE 10-32 Rental Costs by Vehicle Type

The specific vehicles in stock at the rental agency are shown in Figure 10-33.

Data Description

All the data is stored as string types, converted to a numeric type when needed in a computation (such as the cost of daily insurance).

Algorithmic Approach

The algorithmic methods of the program will consist of simple search (for finding and retrieving the requested vehicle information by the user), updating of information (for marking vehicles as reserved or unreserved), and direct calculation (for calculating the total cost of a rental).

Make/Model	Mileage	Num Passengers	Num Doors	Vehicle #
<u>CARS</u>				
Chevrolet Camaro	30 mpg	4	2	WG8JM5492DY
Chevrolet Camaro	30 mpg	4	2	KH4GM4564GD
Ford Fusion	34 mpg	5	4	AB4FG5689GM
Ford Fusion Hybrid	35 mpg	5	4	GH2KL4278TK
Ford Fusion Hybrid	32 mpg	5	4	KU4EG3245RW
Chevrolet Impala	36 mpg	6	4	QD4PK7394JI
Chevrolet Impala	30 mpg	6	4	RK3BM4256YH
Make/Model	Mileage	Num Passengers	Vehicle #	
<u>Vans</u>				
Chrysler Town&Country	25 mpg	7	DK3KG8312UE	
Chrysler Town&Country	25 mpg	7	VM9RE2645TD	
Chrysler Town&Country	25 mpg	7	WK8BF4287DX	
Dodge Caravan	25 mpg	7	QK3FL4278ME	
Dodge Caravan	25 mpg	7	KY8EW2053XT	
Ford Expedition	20 mpg	8	JK2RT8364HY	
Ford Expedition	20 mpg	8	KH4ME4216XW	
Make/Model	Mileage	Cargo Space	Vehicle #	
<u>Trucks</u>				
Ten-Foot	12 mpg	1 bedroom	EJ5KU2435BC	
Ten-Foot	12 mpg	1 bedroom	KF8JP7293EK	
Seventeen-Foot	10 mpg	2 bedrooms	KG4DM5472RK	
Seventeen-Foot	10 mpg	2 bedrooms	PR8JH4893WQ	
Twenty-Four-Foot	8 mpg	4 bedrooms	EP2WR3182QB	
Twenty-Four-Foot	8 mpg	4 bedrooms	TY3GH4290EK	
Twenty-Four-Foot	8 mpg	4 bedrooms	KU9FL4235RH	

FIGURE 10-33 Specific Vehicles of the Vehicle Rental Agency

Overall Program Steps

The overall steps in this program design are given in Figure 10-34.

UML Class Diagram

We give a UML class diagram for the program in Figure 10-35. In addition to the “domain objects” that we have decided on in our analysis, we add a text-based user interface, provided by the RentalAgencyUI class.

Three classes store the information in the system—Vehicle (and its subclasses), VehicleCost, and Reservation. For each of these classes there is a corresponding aggregator class—Vehicles, VehicleCosts, and Reservations—that maintains a collection of the corresponding object type. Each aggregator class has methods for maintaining its collection of objects (for example, addVehicle in the Vehicles class and addVehicleCost in the VehicleCosts class).

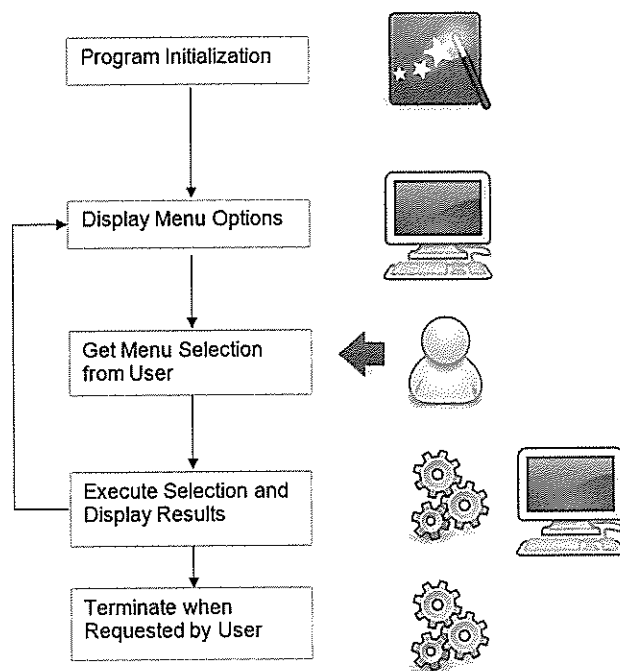


FIGURE 10-34 Overall Design of the Vehicle Rental Agency Program

The Vehicle class has three subclasses—Car, Van, and Truck. It is responsible for maintaining a vehicle's type, its VIN, and its reservation status. Method `getDescription` is provided in the Vehicle class to return the information common to all vehicles: miles per gallon, and a VIN. Each subclass builds on this inherited method to include the specific information for that vehicle type. The Car class stores the maximum number of passengers and number of doors, the Van class stores the maximum number of passengers, and the Truck class stores its length and the number of rooms of storage it can hold.

The VehicleCost class does not have any subclasses. Its `create` (`__init__`) method is passed six arguments: the daily/weekly/weekend rates, the number of free miles, the per mile charge, and the daily insurance rate to initialize the object with. The `getVehicleCost` method of the VehicleCosts aggregating class returns the cost of a specified vehicle type as a single string for display. The Reservation and corresponding Reservations aggregator class are designed in a similar manner.

Finally, a SystemInterface class provides all the methods that any user interface would need for interacting with the system. Such a set of methods is referred to as an API—*Application Programming Interface*. Thus, the SystemInterface object is created first. It then reads all the vehicle rental agency data from text files `VehiclesStock.txt` and `RentalCost.txt` and populates the corresponding objects. Then, an instance of the RentalAgencyUI is created and initialized with a reference to the system interface. The only public method of the

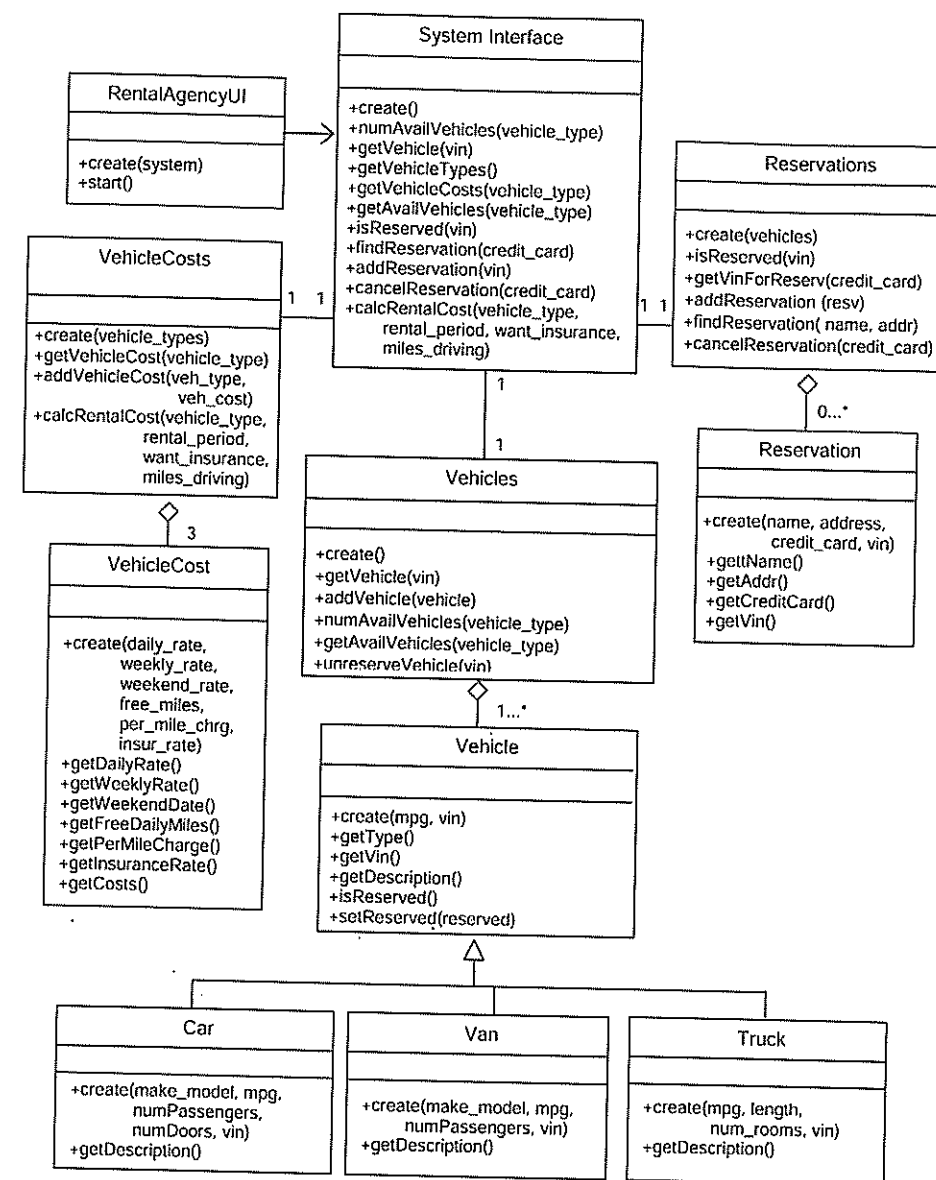


FIGURE 10-35 Class Diagram for Vehicle Rental Agency Program

RentalAgencyUI class, `start`, is called to start the console interaction. The main menu for the program is given in Figure 10-36.

Example use of the system is shown in Figure 10-37. (For the sake of space, the main menu is not repeatedly shown before each selection as in the actual program execution.)

```
<<< MAIN MENU >>>
1 - Display vehicle types
2 - Check rental costs
3 - Check available vehicles
4 - Get cost of specific rental
5 - Make a reservation
6 - Cancel a reservation
7 - Quit

Enter:
```

FIGURE 10-36 Text-Based (Console) Interface for the Vehicle Rental Agency Program

Display Vehicle Types

```
*****
* Welcome to the Friendly Vehicle Rental Agency *
*****

<<< MAIN MENU >>>
1 - Display vehicle types
2 - Check rental costs
3 - Check available vehicles
4 - Get cost of specific rental
5 - Make a reservation
6 - Cancel a reservation
7 - Quit

Enter: 1
----- Types of Vehicles Available for Rent -----

1 - Car
2 - Van
3 - Truck
-----
```

Display Rental Fees for a Given Type Vehicle

```
Enter: 2
Enter type of vehicle
1 - Car
2 - Van
3 - Truck

Enter: 1
----- Rental Charges for Cars -----

      Free   Per Mile   Daily
Daily Weekly Weekend Miles Charge Insurance
24.99 180.00 45.00    100    .15    14.99
-----
```

Check for Available Vehicles

```
Enter: 3
Enter type of vehicle
1 - Car
2 - Van
3 - Truck

Enter: 1
----- Available Cars -----
Chevrolet Camaro passengers: 4 doors: 2 mpg: 30 vin: WG8JM5492DY
Chevrolet Camaro passengers: 4 doors: 2 mpg: 30 vin: KH4GM4564GD
Ford Fusion passengers: 5 doors: 4 mpg: 34 vin: AB4FG5689GH
Ford Fusion Hybrid passengers: 5 doors: 4 mpg: 36 vin: GH2KL4278TK
Ford Fusion Hybrid passengers: 5 doors: 4 mpg: 36 vin: KU4EG3245RW
Chevrolet Impala passengers: 6 doors: 4 mpg: 30 vin: QD4PK7394J1
```

Get the Cost of a Particular Rental

```
Enter: 4
Enter type of vehicle
1 - Car
```

FIGURE 10-37 Example Program Execution (Continued)

```
2 - Van
3 - Truck
Enter: 1
Enter the rental period:
1 - Daily 2 - Weekly 3 - Weekend

Enter: 1
How many days do you need the vehicle? 3
Would you like the insurance? (y/n): n
Number of miles expect to drive?: 240

----- ESTIMATED Car RENTAL COST -----
* You have opted out of the daily insurance *

Daily rental for 3 days would be $ 74.97

Your cost with an estimated mileage of 240 would be 95.97
which includes 100 free miles and a charge of 0.15 per mile
-----

Reserve a Particular Vehicle

Enter: 5

Enter type of vehicle
1 - Car
2 - Van
3 - Truck

Enter: 1
----- Available Cars -----
1-Chevrolet Camaro passengers: 4 doors: 2 mpg: 30 vin: WG8JM5492DY
2-Chevrolet Camaro passengers: 4 doors: 2 mpg: 30 vin: KH4GM4564GD
3-Ford Fusion passengers: 5 doors: 4 mpg: 34 vin: AB4FG5689GH
4-Ford Fusion Hybrid passengers: 5 doors: 4 mpg: 36 vin: GH2KL4278TK
5-Ford Fusion Hybrid passengers: 5 doors: 4 mpg: 36 vin: KU4EG3245RW
6-Chevrolet Impala passengers: 6 doors: 4 mpg: 30 vin: QD4PK7394J1
7-Chevrolet Impala passengers: 6 doors: 4 mpg: 30 vin: RK3BM4256YH

Enter number of vehicle to reserve: 4
Ford Fusion Hybrid passengers: 5 doors: 4 mpg: 36 vin: GH2KL4278TK

Enter first and last name: John Smith
Enter address: 123 Main Street, Sometown 12345
Enter credit card number: 204881605
* Reservation Made *

Cancel a Reservation

Enter: 6
Please enter your credit card number: 204881605

RESERVATION INFORMATION
Name: John Smith 123 Main Street, Sometown 21252
Vehicle: Chevrolet Camaro passengers: 4 doors: 2 mpg: 30 vin: KH4GM4564GD

Confirm Cancellation (y/n): n
* Reservation Cancelled *

Enter: 7
Thank you for using the Friendly Rental Agency
```

FIGURE 10-37 Example Program Execution