

# Class UsedCarLot

java.lang.Object  
UsedCarLot

```
public class UsedCarLot
extends Object
```

This class represents an UsedCarLot object

Author:  
David Yu

## Constructor Summary

Constructors	
Constructor	Description
UsedCarLot()	Instantiates an UsedCarLot object.

## Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	addCar(int indexToAdd, Car carToAdd)	Adds a car to the inventory list at the index specified
void	addCar(Car car)	Adds a car object to the inventory ArrayList
ArrayList <Car>	getInventory()	Returns the inventory, which is an ArrayList of cars
void	moveCar(int indexOfCarToMove, int destinationIndex)	Moves the car located at the specified index to another specified index
Car	sellCarNoShift(int indexOfCarToSell)	Sells the car object located at a specified index.
Car	sellCarShift(int indexOfCarToSell)	Sells the car object located at a specified index
boolean	swap(int index1, int index2)	Swaps the car objects at 2 different indexes.

## Methods inherited from class java.lang.Object

`clone` , `equals` , `finalize` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

## Constructor Details

### UsedCarLot

```
public UsedCarLot()
```

Instantiates an UsedCarLot object.

## Method Details

### getInventory

```
public ArrayList <Car> getInventory()
```

Returns the inventory, which is an ArrayList of cars

**Returns:**

The inventory of car objects

### addCar

```
public void addCar(Car car)
```

Adds a car object to the inventory ArrayList

**Parameters:**

car - A new car object

### swap

```
public boolean swap(int index1,  
                    int index2)
```

Swaps the car objects at 2 different indexes. Returns whether the swap was successful or not

**Parameters:**

index1 - The index of the car you want to swap

index2 - The second index of the car you want to swap

**Returns:**

Whether the swap was successful or not

**addCar**

```
public void addCar(int indexToAdd,  
                  Car carToAdd)
```

Adds a car to the inventory list at the index specified

PRECONDITION:  $0 \leq \text{indexToAdd} < \text{inventory.size()}$

**Parameters:**

indexToAdd - The index where you want to add the car object

carToAdd - The car object that you want to add to the inventory list

**sellCarShift**

```
public Car sellCarShift(int indexOfCarToSell)
```

Sells the car object located at a specified index

PRECONDITION:  $\text{indexOfCarToSell} < \text{inventory.size()}$

**Parameters:**

indexOfCarToSell - The index of the Car object you want to sell

**Returns:**

The car that was sold

**sellCarNoShift**

```
public Car sellCarNoShift(int indexOfCarToSell)
```

Sells the car object located at a specified index.

Replaces the sold car with NULL at the specified index

PRECONDITION:  $\text{indexOfCarToSell} < \text{inventory.size()}$

**Parameters:**

indexOfCarToSell - The index of the Car object you want to sell

**Returns:**

The car that was sold

**moveCar**

```
public void moveCar(int indexOfCarToMove,  
                    int destinationIndex)
```

Moves the car located at the specified index to another specified index

PRECONDITIONS: indexOfCarToMove < inventory.size() destinationIndex < inventory.size()

**Parameters:**

indexOfCarToMove - The index of the Car object you want to move

destinationIndex - The index you want to move the car object to