# Class UsedCarLot

java.lang.Object
    UsedCarLot

---

```
public class UsedCarLot
extends Object
```

This class represents a UsedCarLot object

**Author:**

Katrina Lin

## Constructor Summary

### Constructors

| Constructor | Description |
|---|---|
| `UsedCarLot()` | Instantiates a 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 by indexToAdd |
| void | `addCar(Car addCar)` | Adds the given Car object to the inventory |
| `ArrayList <Car>` | `getInventory()` | Returns the inventory of Car objects |
| void | `moveCar(int indexOfCarToMove, int destinationIndex)` | Moves Car located at index indexOfCarToMove to index destinationIndex |
| Car | `sellCarNoShift (int indexOfCarToSell)` | "sells" the Car located at indexOfCarToSell, REPLACES the Car at indexOfCarToSell with NULL |
| Car | `sellCarShift (int indexOfCarToSell)` | "sells" the Car located at indexOfCarToSell which removes it from the inventory list |
| boolean | `swap(int indice1, int indice2)` | Returns true if the swap of the Car |

| | |
|---|---|
| boolean swap(int indice1, int indice2) | Returns true if the swap of the Car objects at the corresponding indices was successful; returns false otherwise |

### Methods inherited from class java.lang.Object

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

## *Constructor Details*

### UsedCarLot

```
public UsedCarLot()
```

Instantiates a UsedCarLot object

## *Method Details*

### getInventory

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

Returns the inventory of Car objects

**Returns:**

The inventory of Car objects

### addCar

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

Adds the given Car object to the inventory

**Parameters:**

`addCar` - The new Car object

### swap

```
public boolean swap(int indice1,
                    int indice2)
```

Returns true if the swap of the Car objects at the corresponding indices was successful; returns false otherwise

**Parameters:**

`indice1` - The index of the first car to swap

`indice2` - The index of the second car to swap

**Returns:**

true or false depending on whether or not the swap was successful

## addCar

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

Adds a Car to the inventory list at the index specified by indexToAdd

PRECONDITION: 0 <= indexToAdd < inventory.size()

**Parameters:**

`indexToAdd` - The desired index for the Car object to be added at

`carToAdd` - The Car object to be added

## sellCarShift

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

"sells" the Car located at indexOfCarToSell which removes it from the inventory list

Removes it from the inventory list and shifts the remaining Cars in the inventory to the left to fill the gap and reduces the size of inventory by 1

PRECONDITION: indexOfCarToSell < inventory.size()

**Parameters:**

`indexOfCarToSell` - The index of the Car object to be removed from the inventory

**Returns:**

The Car object that was removed from the inventory

## sellCarNoShift

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

"sells" the Car located at indexOfCarToSell, REPLACES the Car at indexOfCarToSell with NULL

Creates an "empty parking spot" on the lot and maintains the size of the inventory

PRECONDITION: indexOfCarToSell < inventory.size()

**Parameters:**

`indexOfCarToSell` - The index of the Car object to be sold and replaced will NULL

**Returns:**

The Car object modified to NULL from the inventory

## moveCar

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

Moves Car located at index indexOfCarToMove to index destinationIndex

If destinationIndex > indexOfCarToMove, moves the Car to right in inventory

If destinationIndex < indexOfCarToMove, moves the Cars to the left in the inventory All other cars in the inventory shift accordingly

PRECONDITIONS: indexOfCarToMove < inventory.size()

destinationIndex < inventory.size()

**Parameters:**

`indexOfCarToMove` - The index of the Car object that will be moved

`destinationIndex` - The destination index at which the Car object will be moved to