

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

dataBaseC

## Class Ticket

java.lang.Object  
dataBaseC.Ticket

```
public class Ticket  
extends java.lang.Object
```

Data structure to hold the ticket and the data associated with it. Used by waiter and chef.

### Author:

cms549

### Field Summary

#### Fields

Modifier and Type	Field and Description
-------------------	-----------------------

int	<b>amountOfDishes</b>
int	<b>amountOfDishesFinished</b>
int	<b>amountOfDishesStarted</b>
int	<b>amountOfDishesUnstarted</b>
boolean	<b>hotFood</b>
java.util.ArrayList<Dish>	<b>listOfDishes</b>
double	<b>price</b>
boolean	<b>recentlySat</b>
char	<b>status</b>

Used to keep track of hot food tables for waiters

List of dishes on the ticket

total price of ticket

Used to keep track of recently sat tables for waiters

status of ticket: u=unstarted, s=semi started, S=started, f=finished

int	<b>tableNumber</b> table number this order belongs to
long	<b>ticketNumber</b> unique id of ticket
long	<b>waiterID</b> id of waiter this ticket belongs to
java.lang.String	<b>waiterName</b> name of waiter this ticket is under

## Constructor Summary

### Constructors

#### Constructor and Description

**Ticket**(java.lang.String waiterName, int tableNum, long waiterID)  
Creates a new empty ticket with the following

## Method Summary

### All Methods    Instance Methods    Concrete Methods

Modifier and Type	Method and Description
void	<b>addDishToTicket</b> (Dish d) Adds dish to ticket and also updates price and status of ticket
boolean	<b>removeDishFromTicket</b> (int indexOfDishInTickList) Removes the dish at index i from the ticket and decrements the price
char	<b>updateStatusOfTicket</b> () Looks through the dishes of the ticket and updates the status of the ticket accordingly

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

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

## Field Detail

**tableNumber**

```
public int tableNumber
```

table number this order belongs to

**waiterName**

```
public java.lang.String waiterName
```

name of waiter this ticket is under

**waiterID**

```
public long waiterID
```

id of waiter this ticket belongs to

**status**

```
public char status
```

status of ticket: u=unstarted, s=semi started, S=started, f=finished

**listOfDishes**

```
public java.util.ArrayList<Dish> listOfDishes
```

List of dishes on the ticket

**ticketNumber**

```
public long ticketNumber
```

unique id of ticket

**price**

```
public double price
```

total price of ticket

**amountOfDishesUnstarted**

```
public int amountOfDishesUnstarted
```

**amountOfDishesStarted**

```
public int amountOfDishesStarted
```

#### **amountOfDishesFinished**

```
public int amountOfDishesFinished
```

#### **amountOfDishes**

```
public int amountOfDishes
```

#### **recentlySat**

```
public boolean recentlySat
```

Used to keep track of recently sat tables for waiters

#### **hotFood**

```
public boolean hotFood
```

Used to keep track of hot food tables for waiters

### ***Constructor Detail***

#### **Ticket**

```
public Ticket(java.lang.String waiterName,  
              int tableNum,  
              long waiterID)
```

Creates a new empty ticket with the following

**Parameters:**

waiterName - - name of waiter for this ticket

tableNum - - table number the ticket is under

waiterID - - id of the waiter

### ***Method Detail***

**addDishToTicket**

```
public void addDishToTicket(Dish d)
```

Adds dish to ticket and also updates price and status of ticket

**Parameters:**

d -

**removeDishFromTicket**

```
public boolean removeDishFromTicket(int indexOfDishInTickList)
```

Removes the dish at index i from the ticket and decrements the price

**Parameters:**

indexOfDishInTickList -

**Returns:****updateStatusOfTicket**

```
public char updateStatusOfTicket()
```

Looks through the dishes of the ticket and updates the status of the ticket accordingly

**Returns:**

old status of ticket

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)