

[Return to Assignments](#)

# DSA - Assignment 1

## Bus Booking System

### using linked list data structure

#### INTRODUCTION

Your **first assignment in this block** will be using linked list data structure for implementing a Bus Booking System (BBS) in Java language.

BBS manages information about Buses, customers and bus booking. These information are:

About a bus:

1. bcode (string): the code of the bus (this should be **unique** for the bus).
2. bus\_name (string): the name of the bus.
3. seat (integer): the number of seats in the bus (**seat > 0**).
4. booked (integer): the number of booked seats in the bus (**booked >= 0** and **booked ≤ seat**).
5. depart\_time (double): The departure time of the bus (**depart\_time >= 0**).
6. arrival\_time (double): The arrival time of the bus (**arrival\_time > depart\_time**).

About a customer:

1. ccode (string): the code of the customer (this should be unique for the customer).
2. cus\_name (string): the name of the customer.
3. phone (string): The phone number of the customer (must contain digits only).

About Booking:

1. bcode (string): the code of the bus to be booked.
2. ccode (string): the code of the customer.
3. seat (integer): the number of seats to be booked on the bus.

#### YOUR TASKS

You should use 3 linked lists, each one is used to store data for buses, customers or bus booking items. You should create linked lists from scratch, do not use list structures available in java like ArrayList, Vector or LinkedList classes.

On running, your program displays the menu as below:

**Bus list** (8 marks):

- 1.1. Load data from file
- 1.2. Input & add to the end
- 1.3. Display data
- 1.4. Save bus list to file
- 1.5. Search by bcode
- 1.6. Delete by bcode
- 1.7. Sort by bcode
- 1.8. Add before position k
- 1.9. Delete the node before the node having bcode = xCode

**Customer list** (1 mark):

- 2.1. Load data from file

- 2.2. Input & add to the end
- 2.3. Display data
- 2.4. Save customer list to file
- 2.5. Search by ccode
- 2.6. Delete by ccode

#### Booking list (1 mark):

- 3.1. Input data
- 3.2. Display data width travel time
- 3.3. Sort by bcode + ccode

### TASKS EXPLANATION

#### Bus list (8 marks):

- 1.1. Load data from file

Allow a user to input the file name that contains information of buses. The content of the text file may be

```
B03 | Sug | 12 | 3 | 11 | 15
B01 | Mil | 10 | 5 | 5.7 | 8
B02 | App | 5 | 2 | 4 | 7
B05 | Roo | 7 | 6 | 15 | 19
B07 | Bee | 11 | 3 | 12 | 13.6
B04 | Boo | 9 | 5 | 5 | 7.1
```

The first line means that: bcode = B03, bus\_name = Sug, seat = 12, booked = 3, depart\_time = 11 and arrival\_time = 15.

The task of this option is to read rows from the text file and **add to the end** of the list. If the list is not empty, then the program asks the user if he/she wants to keep the existing data or not.

- 1.2. Input & add to the head

Allow a user to add new information about a bus. After checking validation of data (including that the bcode could not be duplicated), the bus is added to the head of the list.

- 1.3. Display data

Display data in format:

bcode | bus\_name | seat | booked | depart\_time | arrival\_time | travel\_time  
 where travel\_time = arrival\_time - depart\_time

For example after loading the above file, this option give the output below:

```
bcode|Bus_name|Seat|booked|depart_time|arrival_time|travel_time
-----
--
B03 | Sug | 12 | 3 | 11 | 15 | 4
B01 | Mil | 10 | 5 | 5.7 | 8 | 2.3
B02 | App | 5 | 2 | 4 | 7 | 3
B05 | Roo | 7 | 6 | 15 | 19 | 4
B07 | Bee | 11 | 3 | 12 | 13.6 | 1.6
B04 | Boo | 9 | 5 | 5 | 7.1 | 2.1
```

- 1.4. Save bus list to file

Allow a user to input the file name and save the bus list to the file. The information and format

like the option 1.3.

1.5. Search by bcode

Write the function:

Node search(String xCode) {}

which return reference to the node whose info contains the bus with bcode = xCode.

Allow a user to input the bcode to be searched and display the result: found or not found.

1.6. Delete by bcode

Write the function:

void dele(String xCode) {}

which deletes the node whose info contains the bus with bcode = xCode.

Allow a user to input the bcode to be deleted and then delete the bus having that bcode.

1.7. Sort by bcode

1.8. Add before position k

The position of the first element is 0, the second's is 1

Allow a user to input data for a bus and k, then add the bus before the position k.

1.9. Delete the node before the node having bcode = xCode

Allow a user to input xCode and delete the node before the node having bcode = xCode.

**Customer list** (1 mark):

2.1. Load data from file

Allow a user to input the file name that contains information of customers. The content of the file may be

```
C03 | Hoa | 1902
C01 | La | 1901
C02 | Canh | 1903
C05 | Cay | 1910
```

The first line means that: ccode = C03, name = Hoa, phone = 1902

2.2. Input & add to the end

2.3. Display data

2.4. Save customer list to file

2.5. Search by ccode

2.6. Delete by ccode

**Booking list** (1 mark):

3.1. Input data

Allow a user to input booking item.

When running, the screen looks like:

Enter bus code:

Enter customer code:

Enter number of seats to be booked:

After the user enter bcode and ccode, the program check and acts as follows:

- If bcode not found in the Bus list or ccode not found in the customer list then data is not accepted.

- If both bcode and ccode are found in the booking list then data is not accepted.

- If bcode and ccode found in Buses and customers lists but booked = seat then inform the user that the bus is exhausted.
- If bcode or ccode found and in the Bus list booked < seat and k is the entered seat then if  $k \leq \text{seat} - \text{booked}$  then data is accepted and added to the end of the Booking list.

3.2. Display booking data

3.3. Sort by bcode + ccode

#### Submission Requirements

Create the directory with a name like <class>-<name><roll number>-ASS1, e.g.

**SE0508-QuangTV00456-AS1 (1)**

The (1) directory contains the following files:

1. The run.bat file to run your program.
2. Your source code files.
3. Your input test files (bus.txt and customer.txt).

The statements in run.bat file may be:

```
cls
javac Main.java
java Main
pause
del *.class
```

Compress the folder (1) to .zip (or .rar) file (with the same name) and upload to cms.

#### Assignment assessment

You will be asked to modify immediately and to explain your assignment in class room to be sure that you are really the author of the assignment you submitted.