

## EECS 2011 – Assignment 3

### Description of Program

This Automotive Information System program utilizes a hashmap in order to insert, delete and search for vehicles in an external file. For the first task the user is asked if they want to insert, delete or search for a car in the “autoDatabase”. Each vehicle on the road has a unique VIN number and the user will be asked to enter the vehicles VIN number on completion of the first task. Depending on what the person chose in the first task the program will either search for the vehicle and details in the “autoDatabase” or add/delete a vehicle from the “autoDatabase”. Upon completion the file is rewritten and the program terminates.

```
Welcome to the Automotive Database System!
Enter 'insert', 'delete' or 'search' to proceed:
insert
Enter VIN:
12JY321D
Enter auto information in one line ('manf' 'model' 'year' 'colour'):
Toyota Camry 2018 Silver
|
```

### Robustness of Program

When the user uses the program, they will be given information on usage every step of the way. If the user enters an invalid option when picking to insert, delete or search they will be prompted with “Invalid Option!” and the program will terminate. Furthermore, the fileWriter and fileReader are setup to write the file in a certain format and read the file in a certain format to ensure consistency. In order to make sure no one tampers with the data in the “autoDatabase” file, upon completion of rewriting the file it is set to read-only so that it cannot be changed to the improper format.

```
file.setWritable(true);
writeMap(map, "autoDatabase");
file.setReadOnly();
```

### How the program works?

The program uses the java hashmap implementation to take the data from the “autoDatabase” and add it into a hashmap. When the program is done running, the data from the hashmap is written to the “autoDatabase”.

```
BufferedReader reader = new BufferedReader(new FileReader(fileName));


String line;
while ((line = reader.readLine()) != null) {
    String parts[] = line.split(" ");
    map.put(parts[0], parts[1]);
}
reader.close();
```

### Runtime and Space Usage

In the worst case, the insertion, deletion and search of the hashmap in this program will take  $O(n)$  time. This only occurs when all the keys inserted into the map collide and need to be sorted. However, since VIN numbers for vehicles are all unique, the user should never have this problem so the run time may be faster. Each operation of the hashmap is  $O(1)$ . The total space the hashmap uses is  $O(n)$ .

## Testing the program

This is the initial “autoDatabase” file.


 autoDatabase - Notepad  
File Edit Format View Help  
1HGBH41 Toyota Camry 2018 Silver  
1G32Z21 Honda Accord 2006 Blue  
23J1Z32 Infiniti G35 2003 Silver  
32H2AV9 Toyota Supra 1998 Black  
5AB23T4 BMW 330i 2007 Black

Inserting new entry.

---

```
Welcome to the Automotive Database System!
Enter 'insert', 'delete' or 'search' to proceed:
insert
Enter VIN:
2J4R5NT1
Enter auto information in one line ('manf' 'model' 'year' 'colour'):
Ferrari F40 1987 Red
.
```

Result.

 autoDatabase - Notepad  
File Edit Format View Help  
1HGBH41 Toyota Camry 2018 Silver  
1G32Z21 Honda Accord 2006 Blue  
23J1Z32 Infiniti G35 2003 Silver  
2J4R5NT1 Ferrari F40 1987 Red  
32H2AV9 Toyota Supra 1998 Black  
5AB23T4 BMW 330i 2007 Black  
.

Deleting 1HGBH41 key.

```
Welcome to the Automotive Database System!
Enter 'insert', 'delete' or 'search' to proceed:
delete
Enter VIN:
1HGBH41
.
```

Result.



autoDatabase - Notepad

File Edit Format View Help

```
1G32Z21 Honda Accord 2006 Blue
23J1Z32 Infiniti G35 2003 Silver
2J4R5NT1 Ferrari F40 1987 Red
32H2AV9 Toyota Supra 1998 Black
5AB23T4 BMW 330i 2007 Black
```

Searching for 1G32Z21 key.

Welcome to the Automotive Database System!

Enter 'insert', 'delete' or 'search' to proceed:

search

Enter VIN:

1G32Z21

Honda