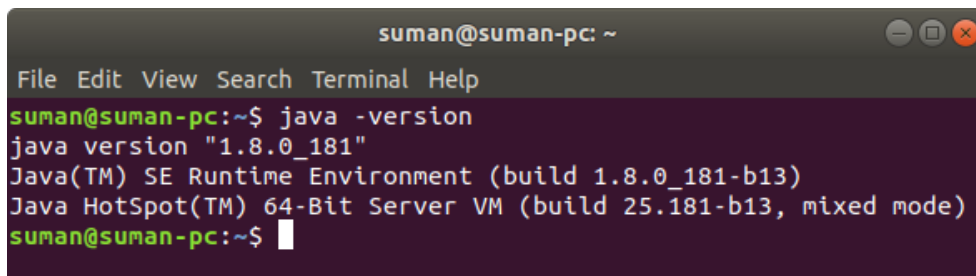


Developer: Suman Barua
~: Programming Exercise :~
Routing of Telephone Calls – Developed with JAVA

I developed this project as per given instruction manual. Some necessary information given below -

Programming Language: JAVA

Java Version:

A screenshot of a terminal window titled 'suman@suman-pc: ~'. The terminal shows the command 'java -version' being executed. The output is: 'java version "1.8.0_181"', 'Java(TM) SE Runtime Environment (build 1.8.0_181-b13)', and 'Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)'. The prompt 'suman@suman-pc:~\$' is visible at the bottom.

```
suman@suman-pc: ~  
File Edit View Search Terminal Help  
suman@suman-pc:~$ java -version  
java version "1.8.0_181"  
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)  
suman@suman-pc:~$
```

Used IDE:

I have developed, executed and tested the project using *NetBeans IDE 8.2*. I would recommend the same environment for testing.

Operating System: Ubuntu 18.04.1 LTS

Project Directory Structure:

Root Directory – ProgrammingExercise

Script File Path -

ProgrammingExercise => src => programmingexercise => ProgrammingExercise.java

Testing Instruction:

Initially two operators (Operator-A & Operator-B) are given. If you want to test the script with more operators' data (Operator-C & Operator-D), then please uncomment the following commented code lines of function: *getTelephoneOperators()*.

```

22 // Uncomment Operator-C and Operator-D if you want to check with more operators
23 // // Initialize Operator-C
24 // Hashtable<String, Double> OperatorC = new Hashtable<String, Double>();
25 // OperatorC.put("1", 2.0);
26 // OperatorC.put("44", 1.0);
27 // OperatorC.put("46", 3.0);
28 // OperatorC.put("46725", 4.0);
29 // OperatorC.put("467", 1.2);
30 // OperatorC.put("4672", 2.0);
31 //
32 // // Initialize Operator-D
33 // Hashtable<String, Double> OperatorD = new Hashtable<String, Double>();
34 // OperatorD.put("46", 4.0);
35 // OperatorD.put("467", 2.0);
36 // OperatorD.put("46725", 3.0);
37 // OperatorD.put("4672", 1.0);
38 // OperatorD.put("48", 3.0);
39
40 // Serialized the price-lists in the Hashtable as per operator's name
41 Hashtable<String, Hashtable> operators = new Hashtable<String, Hashtable>();
42 operators.put("Operator-A", OperatorA); // Adding Operator-A to HashTable
43 operators.put("Operator-B", OperatorB); // Adding Operator-B to HashTable
44
45 // Uncomment Operator-C and Operator-D if you want to check with more operators
46 // operators.put("Operator-C", OperatorC); // Adding Operator-C to HashTable
47 // operators.put("Operator-D", OperatorD); // Adding Operator-D to HashTable
48

```