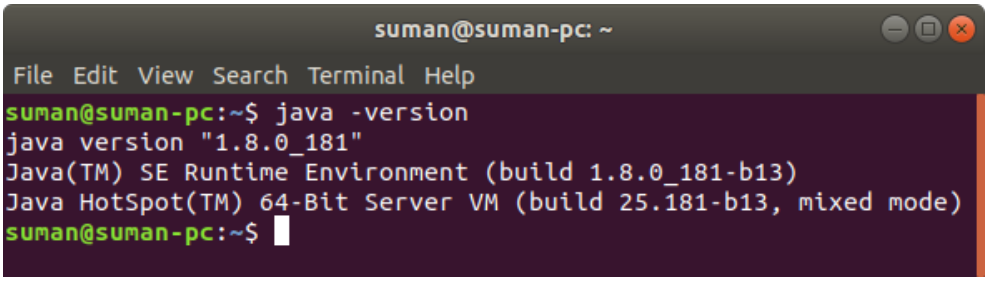
Developer: Suman Barua
~: Programming Exercise :~
Routing of Telephone Calls – Developed with JAVA

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

Programming Language: JAVA

Java Version:



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()*.

```
        // Uncomment Operator-C and Operator-D if you want to check with more operators
//        // Initialize Operator-C
//        Hashtable<String, Double> OperatorC = new Hashtable<String, Double>();
//        OperatorC.put("1", 2.0);
//        OperatorC.put("44", 1.0);
//        OperatorC.put("46", 3.0);
//        OperatorC.put("46725", 4.0);
//        OperatorC.put("467", 1.2);
//        OperatorC.put("4672", 2.0);
//
//        // Initialize Operator-D
//        Hashtable<String, Double> OperatorD = new Hashtable<String, Double>();
//        OperatorD.put("46", 4.0);
//        OperatorD.put("467", 2.0);
//        OperatorD.put("46725", 3.0);
//        OperatorD.put("4672", 1.0);
//        OperatorD.put("48", 3.0);

        // Serialized the price-lists in the Hashtable as per operator's name
        Hashtable<String, Hashtable> operators = new Hashtable<String, Hashtable>();
        operators.put("Operator-A", OperatorA); // Adding Operator-A to HashTable
        operators.put("Operator-B", OperatorB); // Adding Operator-B to HashTable

        // Uncomment Operator-C and Operator-D if you want to check with more operators
//        operators.put("Operator-C", OperatorC); // Adding Operator-C to HashTable
//        operators.put("Operator-D", OperatorD); // Adding Operator-D to HashTable
```