



Read Me:-

- i. Before going through below exercises please visit the link given below, where you can experience the coding standard that each and every developer should follow.
- ii. This Code Conventions for the Java Programming Language document contains the standard conventions that Sun follow and recommend that we should follow. It covers filenames, file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices and includes a code example.
- iii. LINK - <http://www.oracle.com/technetwork/java/codeconv-138413.html>

Collection Framework Lab Exercise Day-2

Duration: 2 Hours

Program: 5

Write a java program to convert a given array of Strings into a Map.

A ArrayToMap class is given to you. Add the following method in the class:

Implement the static method `convertToMap(String[] names)`

This method must read the given array of names and insert them one by one into a `Map<Integer,String>` and returns this map as the output.

Each name in the array is the value in the map. Insert Integer numbers as the key.

The key in the map must start from 1 and be continuous without missing numbers in between.

If the array is null, return null. If the array is empty, return an empty map.

If the array contains null elements in between, ignore that element and add the non-null strings into the map. Do not insert a null value in the map.

Program: 6

A class `MapCharacterOccurrence` is given to you. Implement the following features in the class.

Implement the method `getOccurrences(String)`: this method should count each character occurrences in the given string and return it as a Map.

Here store each character as key, and its occurrences count as value.

Spaces should be treated as underscore (`_`)

Count the Occurrence of Upper case and lower case characters separately.

If the string is empty, then return null

Ex:- input: Hello World

output: {H=1, e=1, l=3, o=2, _=1, W=1, r=1, d=1}

Hint : create Reference Of `LinkedHashMap`.

A class `Tester` is given to you with a main method. Create `MapCharacterOccurrence` objects here and test its methods with various inputs.

Program: 7

Three classes are given to you,

CabCustomer

CabCustomerService

CabCustomerServiceTester

Define the following in the class CabCustomer

private : custId int, customerName String, pickupLocation String, dropLocation String, distance int, phone String

Generate Getter/Setter for the all fields.

Implement the default constructor.

Implement/Generate the parameterized constructor in the order as defined above, i.e., custId, Name, Pickup, Drop, Distance, Phone

Define the following in the class CabCustomerService and write logics in the following methods:

private : Generic ArrayList to represent list of CabCustomers.

public : addCabCustomer() : Add the customer object parameter to the ArrayList

isFirstCustomer(): Check whether the customer object parameter is already existing in arrayList.

Note : If phone number of a customer matches with any of the phone numbers of the array list, then consider it as a existing customer, otherwise consider the customer as new customer.

calculateBill() should calculate and return the customer bill based on following rules

1) if the customer is new return 0;

2) if the customer's travel distance is below or equal to 4 kms then return 80 (Rs).

3) if the customer travel distance is above 4 kms calculate bill 80 + 6 per each km.

Ex: Any distance for new customer return 0;

distance 4 return 80

distance 6 return 80 + 6 * 6;

printBill() which should return the bill of the customer object parameter in the following format:

output : JOHN Please pay your bill of Rs.0.0

SMITH Please pay your bill of Rs.180.0

Note :

Assume one customer books only one cab at a time.

No charge for customer booking the cab for the first time.

Customer's phone number is key to test a new customer or old customer.

Distance should be treated as kilometers

A class CabCustomerServiceTester is given with main() where you can create various objects and test them.

All the Best