

**~~Exercise 1~~**

~~Download the code UsingMaps.java and run it step by step by removing the comments.~~

~~The aim is to understand how the TreeMap methods are implemented and what they do.~~

~~Write code to implement the TODOs in the comments.~~

**Q1**

In the mapTreeMap(() method, code the following:

* Declare a reference of type Map that is generically typed for keys of type String and values of type Integer. Name the reference ‘map’. The implementation type is able **to sort itself**! Use type inference (the diamond operator).
* Insert, in this order, the following into the map:
  + “Ford” is 2019
  + “Audi” is 2015
  + “Polo” is 2023
  + “Beetle” is 2012
* Output the map. You should get: {Audi=2015, Beetle=2012, Ford=2019, Polo=2023}
  + note that the map is sorted automatically by its natural ordering of keys i.e. alphabetic order for the names.
* Using the API, output whether the map contains “Ford” as a key. This should output true.
* Using the API, output whether the map contains 2023 as a value. This should output true also.
* Using the API, remove “Polo” from the map.
* Using the API, change “Audi” age to 2016.
* Output the map. You should get: {Audi=2016, Beetle=2012, Ford=2019}
* Using the API and an enhanced-for loop, output the keys in the map. You should get Audi, Beetle, Ford in that order.
* Using the API and an enhanced-for loop, output the values in the map. You should get 2016, 2012, 2019 in that order.

**Q2**

In the mapHashMap() method, code the following:

* Declare a reference of type Map that is generically typed for keys of type String and values of type Integer. Name the reference ‘map’. Use type inference (the diamond operator).
* Insert, in this order, the following into the map:
  + “Ford” is 2019
  + “Audi” is 2015
  + “Polo” is 2023
  + “Beetle” is 2012
* Output the map. You should get the key value pairs but the order will be different e.g. { Polo=2023,Audi=2015, Beetle=2012, Ford=2019 }
* Using the API, output whether the map contains “Ford” as a key. This should output true.
* Using the API, output whether the map contains 2023 as a value. This should output true also.
* Using the API, remove “Polo” from the map.
* Using the API, change “Audi” age to 2016.
* Output the map. Note that the map entries are not sorted.

**Q3. Revision**

Code the following:

In main() call the method exceptions1()

In the method exceptions1(), call the method exceptions2().

In the method exceptions2(), throw an IOException. This exception is caught and handled in the method main().

**Q4. Revision**

Code the listOfFruits method as follows:

* Declare a reference of type List that is generically typed for String i.e. it will only hold String references. Name the reference ‘list’. The implementation type is ArrayList. Use type inference (the diamond operator).
* Insert the following in order: “Apple”, “Banana”, “Cherry”, “Elderberry”,”Fig”
* Using the API i.e. do not hardcode the index, insert “Damson” in its correct position alphabetically. In other words, when you output the list, you should get [“Apple”, “Banana”, “Cherry”, “Damson”,”Elderberry”,”Fig”].
* Using the API, output the size of the list. This should be 6.
* Using the API, output whether the list contains “Pear”. This should output false.
* Using the API, delete “Elderberry” from the list.
* Output the list. You should get [“Apple”, “Banana”, “Cherry”, “Damson”,”Fig”].
* Using the API, output if the list is empty or not. This should output false.
* Using the API, change the “Fig” to a “Cherry”.
* Output the list. You should get [“Apple”, “Banana”, “Cherry”, “Damson”,”Cherry”].