

Implement the UML and all the following questions in one file.

**Q1.**

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 exceptions1 ().

**Q2**

Code the following in the method set():

* Declare a reference of type Set that is generically typed for String i.e. it will only hold String references. Name the reference ‘set’. The implementation type is able to sort itself! Use type inference (the diamond operator).
* Insert the following in this order: “A”, “B”, “C”, “D”.
* Output the set. You should get [A, B, C, D].
* Using the API, insert “F”.
* Using the API, delete “C” from the set.
* Output the set. You should get [A, B, D, F].
* Using the API, output if the set contains the string “E”. This should output false.
* Using the API, remove all entries.
* Using the API, output if the set is empty. This should output true.

**Q3**

In the map() 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.

**Q4.**

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”].