**Exercises on Collections**

In the Customer Project:

* In the **CustomerContainer** class**,** change the array to a new object of type TreeSet , using a comparator object to keep the container sorted by customers’ names. Change all the methods properly. Print the customer list.
* After a little while, we are asked not to enable more than one customer with the same name – If such a customer already exists, print a suitable message, else print the details of the customer added. Use the HashSet’s capabilities for this behavior.

Write a program BookSearch that performs the following tasks using collections:

1. It counts the number of unique words in a large text file (use any long text file from the web).

To read a text file:

**Scanner input = new Scanner(new File("bookFile.txt"));**

**while (input.hasNext()) {**

**String word = input.next();**

**. . .**

**}**

1. Allow the user to type a word and report if it appears on the book or not.
2. Allow the user to type a word and report the number of times that word appeared in the book.
3. Modify the word count program to print **every word** that appeared in the book **at least 5** times, in sorted order from least to most occurrences.
4. Modify the Book Search program to eliminate any words that are plural or all-uppercase

in the collection.

1. We want to create an index for the book, so first we need to add to any page in the book it’s page number , using Notepad or any other text editor.
2. Print an index for the book, containing each word in alphabetical order and the list of every page in which the word appears separated by ‘,’ ( the pages numbers must be unique and in ascending order)