

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/javase/documentation/codeconventions-135099.html>

COLLECTIONS LAB EXERCISE STAGE-1

1. Array to Collection-Converting values of an array into collection...
 - a. *List interface is a member of the Java Collection Framework.*
 - b. Use `asList()` method of **Array** class and pass the array as argument to the `asList()` method.
 - c. Use **Iterator** to get all the elements present in the list object.
 2. Storing and Retrieving Objects data to and from collection object...
 - a. In a class room 10 students write an exam.
 - b. Store all student information as an object into List implemented class object.
 - c. Display the students names and their marks who got
 - i. Highest marks in the classroom.
 - ii. Least marks in the classroom
 3. Insert all elements of other Collection to Specified Index of ArrayList...
 - a. Write a program to create a Main class
 - b. Create the **ArrayList** object like
`ArrayList<String> arrayList = new ArrayList<String>();`
 - c. Add elements to **ArrayList** object
 - d. Create Vector object and add elements to vector object
 - e. Insert all elements of **Vector** to **ArrayList** at index 1
 - f. Using 'addAll()' method of **ArrayList** class add display all elements present in the **ArrayList** object using for-each loop.
-

4. Get Enumeration over ArrayList...
 - a. Create a class **GetEnumerationOverArrayList** which implements *main* method inside main method create an **ArrayList** object
 - b. Add some elements to **ArrayList** object.
 - c. Get **Enumeration** by calling enumeration method on Collections object by passing **ArrayList** object as argument to that method and print all the elements present in the **ArrayList** object using *hasMoreElements()* and *nextElement()* methods.
5. Search collection element...
 - a. Create a class **SearchCollectionElement** which implements *main* method inside main method create a **LinkedList** object.
 - b. Create an object of **DateFormatSymbols** get the all the months by calling *getMonths()* method on the **DateFormatSymbols** object and assign them to String array.
 - c. Add all months to **LinkedList** object using a loop
 - d. Sort the elements present in the **LinkedList**.
 - e. Display the elements present in the **LinkedList** object.
 - f. Search a particular month using *binarySearch()* method and display the position of the month.
4. Store and retrieve from Set implemented Classes...
 - a. Create a class named **Student** with properties *studentName*, *mark*, *dob*, and *college*.
 - b. Implement *getXXX()* and *setXXX ()* methods.
 - c. Now create a class with name *StuentSetDemo* and implements *main()* method.
 - d. In *main()* method create an object of **HashSet** with *initialCapacity* and *loadFactor* class.
 - e. Create and add some **Student** objects into this object.
4. Implementation of **Comparable** on user defined object.
 - a. Create a class *employee* which contains the fields *employeeid*, *employee firstname*, *employee lastname*, *salary* and *experience*
 - b. Create 10 *employee* objects and assign the values to the fields.
 - c. Sort the 10 *employee* objects based on their first name
 - d. If any two *employees* first name is equal sort based on the last name if the *employees* last name also equal sort that *employee* objects based on their salary or experience.
 - e. Sort the *employee* objects based on their *employee id* (ascending order)
5. Arrange the *employees* in the reverse order according their *employeeid* [use reverse order method]

6. Implementation of Comparator on user defined object.
- a. Create a student class with properties id, math's, physics & chemistry.
 - b. Create 10 objects of Student class and sort those Student objects based on their id.
 - c. If two students id are same sort that id's based on Math's marks.
 - d. If two students Math's marks are same sort them based on physics marks.

ALL THE BEST
