BCA1317A02 Java Programming Lab

Lab #4 (Due date: 02/10/2020 11:59 PM)

Topics: Collection Frameworks

ArrayList Exercises

--

1. Write a Java program to create a new array list, add some colors (string) and print out the collection.
See the solution

2. Write a Java program to iterate through all elements in a array list.
See the solution

3. Write a Java program to insert an element into the array list at the first position.
See the solution

4. Write a Java program to retrieve an element (at a specified index) from a given array list.
See the solution

5. Write a Java program to update specific array element by given element.
See the solution

6. Write a Java program to remove the third element from a array list.
See the solution

7. Write a Java program to search an element in a array list.
See the solution

8. Write a Java program to sort a given array list.
See the solution

9. Write a Java program to copy one array list into another.
See the solution

10. Write a Java program to shuffle elements in a array list.
See the solution

11. Write a Java program to reverse elements in a array list.
See the solution

12. Write a Java program to extract a portion of a array list.
See the solution

13. Write a Java program to compare two array lists.
See the solution

14. Write a Java program of swap two elements in an array list.
See the solution

15. Write a Java program to join two array lists.
See the solution

16. Write a Java program to clone an array list to another array list.
See the solution

17. Write a Java program to empty an array list.
See the solution

18. Write a Java program to test an array list is empty or not.
See the solution

19. Write a Java program to trim the capacity of an array list the current list size.
See the solution

20. Write a Java program to increase the size of an array list.
See the solution

21. Write a Java program to replace the second element of a ArrayList with the specified element.
See the solution

22. Write a Java program to print all the elements of a ArrayList using the position of the elements.
See the solution

LinkedList Exercises

--

1. Write a Java program to append the specified element to the end of a linked list.
See the solution

2. Write a Java program to iterate through all elements in a linked list.
See the solution

3. Write a Java program to iterate through all elements in a linked list starting at the specified position.
See the solution

4. Write a Java program to iterate a linked list in reverse order.
See the solution

5. Write a Java program to insert the specified element at the specified position in the linked list.
See the solution

6. Write a Java program to insert elements into the linked list at the first and last position.
See the solution

7. Write a Java program to insert the specified element at the front of a linked list.
See the solution

8. Write a Java program to insert the specified element at the end of a linked list.
See the solution

9. Write a Java program to insert some elements at the specified position into a linked list.
See the solution

10. Write a Java program to get the first and last occurrence of the specified elements in a linked list.
See the solution

11. Write a Java program to display the elements and their positions in a linked list.
See the solution

12. Write a Java program to remove a specified element from a linked list.
See the solution

13. Write a Java program to remove first and last element from a linked list.
See the solution

14. Write a Java program to remove all the elements from a linked list.
See the solution

15. Write a Java program of swap two elements in an linked list.
See the solution

16. Write a Java program to shuffle the elements in a linked list.
See the solution

17. Write a Java program to join two linked lists.
See the solution

18. Write a Java program to clone an linked list to another linked list.
See the solution

19. Write a Java program to remove and return the first element of a linked list.
See the solution

20. Write a Java program to retrieve but does not remove, the first element of a linked list.
See the solution

21. Write a Java program to retrieve but does not remove, the last element of a linked list.
See the solution

22. Write a Java program to check if a particular element exists in a linked list.
See the solution

23. Write a Java program to convert a linked list to array list.
See the solution

24. Write a Java program to compare two linked lists.
See the solution

25. Write a Java program to test an linked list is empty or not.
See the solution

26. Write a Java program to replace an element in a linked list.
See the solution

HashSet Exercises

1. Write a Java program to append the specified element to the end of a hash set.
See the solution

2. Write a Java program to iterate through all elements in a hash list.
See the solution

3. Write a Java program to get the number of elements in a hash set.
See the solution

4. Write a Java program to empty an hash set.
See the solution

5. Write a Java program to test a hash set is empty or not.
See the solution

6. Write a Java program to clone a hash set to another hash set.
See the solution

7. Write a Java program to convert a hash set to an array.
See the solution

8. Write a Java program to convert a hash set to a tree set.
See the solution

9. Write a Java program to convert a hash set to a List/ArrayList.
See the solution

10. Write a Java program to compare two hash set.
See the solution

11. Write a Java program to compare two sets and retain elements which are same on both sets.
See the solution

12. Write a Java program to remove all of the elements from a hash set.
See the solution

TreeSet Exercises

--

1. Write a Java program to create a new tree set, add some colors (string) and print out the tree set.
See the solution

2. Write a Java program to iterate through all elements in a tree set.
See the solution

3. Write a Java program to add all the elements of a specified tree set to another tree set.
See the solution

4. Write a Java program to create a reverse order view of the elements contained in a given tree set.
See the solution

5. Write a Java program to get the first and last elements in a tree set.
See the solution

6. Write a Java program to clone a tree set list to another tree set.
See the solution

7. Write a Java program to get the number of elements in a tree set.
See the solution

8. Write a Java program to compare two tree sets.
See the solution

9. Write a Java program to find the numbers less than 7 in a tree set.
See the solution

10. Write a Java program to get the element in a tree set which is greater than or equal to the given element.
See the solution

11. Write a Java program to get the element in a tree set which is less than or equal to the given element.
See the solution

12. Write a Java program to get the element in a tree set which is strictly greater than or equal to the given element.
See the solution

13. Write a Java program to get an element in a tree set which is strictly less than the given element.
See the solution

14. Write a Java program to retrieve and remove the first element of a tree set.
See the solution

15. Write a Java program to retrieve and remove the last element of a tree set.
See the solution

16. Write a Java program to remove a given element from a tree set.
See the solution

PriorityQueue Exercises

1. Write a Java program to create a new priority queue, add some colors (string) and print out the elements of the priority queue.
See the solution

2. Write a Java program to iterate through all elements in priority queue.
See the solution

3. Write a Java program to add all the elements of a priority queue to another priority queue.
See the solution

4. Write a Java program to insert a given element into a priority queue.
See the solution

5. Write a Java program to remove all the elements from a priority queue.
See the solution

6. Write a Java program to count the number of elements in a priority queue.
See the solution

7. Write a Java program to compare two priority queues.
See the solution

8. Write a Java program to retrieve the first element of the priority queue.
See the solution

9. Write a Java program to retrieve and remove the first element.
See the solution

10. Write a Java program to convert a priority queue to an array containing all of the elements of the queue.
See the solution

11. Write a Java program to convert a Priority Queue elements to a string representation.
See the solution

12. Write a Java program to change priorityQueue to maximum priorityqueue.
See the solution

HashMap Exercises

--

1. Write a Java program to associate the specified value with the specified key in a HashMap.
See the solution

2. Write a Java program to count the number of key-value (size) mappings in a map.
See the solution

3. Write a Java program to copy all of the mappings from the specified map to another map.
See the solution

4. Write a Java program to remove all of the mappings from a map.
See the solution

5. Write a Java program to check whether a map contains key-value mappings (empty) or not.
See the solution

6. Write a Java program to get a shallow copy of a HashMap instance.
See the solution

7. Write a Java program to test if a map contains a mapping for the specified key.
See the solution

8. Write a Java program to test if a map contains a mapping for the specified value.
See the solution

9. Write a Java program to create a set view of the mappings contained in a map.
See the solution

10. Write a Java program to get the value of a specified key in a map.
See the solution

11. Write a Java program to get a set view of the keys contained in this map.
See the solution

12. Write a Java program to get a collection view of the values contained in this map.
See the solution

TreeMap Exercises

--

1. Write a Java program to associate the specified value with the specified key in a Tree Map.
See the solution

2. Write a Java program to copy a Tree Map content to another Tree Map.
See the solution

3. Write a Java program to search a key in a Tree Map.
See the solution

4. Write a Java program to search a value in a Tree Map.
See the solution

5. Write a Java program to get all keys from the given a Tree Map.
See the solution

6. Write a Java program to delete all elements from a given Tree Map.
See the solution

7. Write a Java program to sort keys in Tree Map by using comparator.
See the solution

8. Write a Java program to get a key-value mapping associated with the greatest key and the least key in a map.
See the solution

9. Write a Java program to get the first (lowest) key and the last (highest) key currently in a map.
See the solution

10. Write a Java program to get a reverse order view of the keys contained in a given map.
See the solution

11. Write a Java program to get a key-value mapping associated with the greatest key less than or equal to the given key.
See the solution

12. Write a Java program to get the greatest key less than or equal to the given key.
See the solution

13. Write a Java program to get the portion of a map whose keys are strictly less than a given key.
See the solution

14. Write a Java program to get the portion of this map whose keys are less than (or equal to, if inclusive is true) a given key.
See the solution

15. Write a Java program to get the least key strictly greater than the given key. Return null if there is no such key.
See the solution

16. Write a Java program to get a key-value mapping associated with the greatest key strictly less than the given key. Return null if there is no such key.
See the solution

17. Write a Java program to get the greatest key strictly less than the given key. Return null if there is no such key.
See the solution

18. Write a Java program to get NavigableSet view of the keys contained in a map.
See the solution

19. Write a Java program to remove and get a key-value mapping associated with the least key in a map.
See the solution

20. Write a Java program to remove and get a key-value mapping associated with the greatest key in this map.
See the solution

21. Write a Java program to get the portion of a map whose keys range from a given key (inclusive), to another key (exclusive).
See the solution

22. Write a Java program to get the portion of a map whose keys range from a given key to another key.
See the solution

23. Write a Java program to get a portion of a map whose keys are greater than or equal to a given key.
See the solution

24. Write a Java program to get a portion of a map whose keys are greater than to a given key.
See the solution

25. Write a Java program to get a key-value mapping associated with the least key greater than or equal to the given key. Return null if there is no such key.
See the solution

26. Write a Java program to get the least key greater than or equal to the given key. Returns null if there is no such key.
See the solution

Quick Reference

https://en.wikiversity.org/wiki/Java_Collections_Overview

https://docs.oracle.com/javase/tutorial/collections/interfaces/collection.html

https://howtodoinjava.com/java-collections/