

Explore More

Subscription : Premium CDAC NOTES & MATERIAL @99



Contact to Join
Premium Group



Click to Join
Telegram Group

<CODEWITHARRAY'S/>

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

freelance_Project available to buy contact on 8007592194		
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql
51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=ILOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FlzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/lynlouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyl4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizlyJlh
15	Employee management system project	https://youtu.be/ID1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSIsm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

MCQ on Collections In Java

Q#1. Which of the following statement(s) is/are correct about the List collection?

- (a) Just like an array, List indexes start from 0.
- (b) List interface has got many default methods in Java 8.
- (c) List allows you to have 'null' elements.
- (d) List doesn't support Generics.

Answer: (a), (b), (c)

Explanation: The List interface indexes start from 0, allows null elements & also supports Generics. It allows duplicate elements and maintains an ordered collection. Furthermore, List interface has got many default methods in Java 8.

Q#2. What is the correct way to create an ArrayList of Strings?

- (a) `ArrayList<String> list = new ArrayList[];`
- (b) `ArrayList list = new List<String>();`
- (c) `ArrayList list = new ArrayList<String>();`
- (d) `ArrayList<String> list = new ArrayList<>();`

Answer: (d) `ArrayList<String> list = new ArrayList<>();`

Explanation: While creating an ArrayList of a specific type, the variable declaration should include the type.

Q#3. Which collection allows you to retrieve elements in the order they were inserted?

- (a) `ArrayList`
- (b) `LinkedList`
- (c) `LinkedHashMap`
- (d) `LinkedHashSet`

Answer: (a), (b), (c), (d)

Explanation: The LinkedList & ArrayList classes implement the List interface and allows us to retrieve elements based on the insertion order. LinkedHashMap is a map implementation that maintains the insertion order of key-value pairs. Similarly, the LinkedHashSet is a set implementation that maintains the insertion order of elements.

Q#4. What is the output of the following code?

```
ArrayList<String> list = new ArrayList<>();  
list.add("Java");  
list.add("Python");  
System.out.println(list.remove(0));
```

- (a) Java
- (b) Python
- (c) true
- (d) Compilation error

Answer: a) Java

Explanation: The remove() method is used to remove an element at the specified index in an ArrayList. In this case, it removes "Java" at index 0 and returns it.

Q#5. Which of the following is incorrect statement about the Hashtable class?

- (a) Hashtable class implements Map interface.
- (b) Hashtable class extends the Dictionary class.
- (c) Hashtable class implements Cloneable interface.
- (d) Hashtable class extends Properties class.

Answer: d) Hashtable class extends Properties class.

Explanation: Hashtable class doesn't extend Properties class. Alternatively, the Properties class extends Hashtable class.

Q#6. Which of the following collection class/(s) is/are synchronized and thread-safe?

- (a) Stack
- (b) Hashtable
- (c) Properties
- (d) Vector

Answer: (a), (b), (c), (d)

Explanation: All of the above classes are synchronized and thread safe in java.

Q#7. Which one is the correct way to iterate over elements in an ArrayList?

- (a) `for (int i = 0; i < list.size(); i++)`
- (b) `for (int i : list)`
- (c) `for (Object obj : list)`
- (d) `for (Iterator it = list.iterator(); it.hasNext();)`

Answer: c) `for (Object obj : list)`

Explanation: The correct way to iterate over elements in an ArrayList is by using a for-each loop and declaring the loop variable as the type of the elements in the list.

Q#8. Which of the following statement is incorrect about Deque interface in Java?

- (a) It is typically pronounced as "deck".
- (b) It stands for doubly-ended queue.
- (c) It is mostly used to implement some algorithms and data structures.
- (d) It can either be used as a queue(FIFO) or as a stack(LIFO).

Answer: b) It stands for doubly-ended queue.

Explanation: The [Deque](#) interface extends the Collection interface and adds support for inserting elements at both the ends. It stands for "double-ended queue", not the doubly-ended queue. All

other statements are correct about Deque. You may also read [When to use Deque](#) in some more detail.

Q#9. Suppose you are working in a Java project. You have a requirement where you need a dynamic or resizable array-like data structure that can permit you to add, remove, and access elements at a specific index efficiently. It should also automatically adjust its size when the elements are added or removed. Which of the following class will you use to satisfy your requirement?

- (a) Vector
- (b) LinkedList
- (c) ArrayList
- (d) HashMap

Answer: c) ArrayList

Explanation: The ArrayList class provides a resizable array implementation of the List interface. It dynamically grows and shrinks as elements are added or removed.

Q#10. Which collection class provides a way to store elements in key-value pairs with no duplicate keys?

- (a) ArrayList
- (b) Hashtable
- (c) TreeMap
- (d) LinkedHashMap

Answer: c) TreeMap

Explanation: The TreeMap class provides a way to store elements in key-value pairs with no duplicate keys. It orders the keys in ascending order by default.

Q#11. What is the output of the following code?

```
LinkedList<String> list = new LinkedList<>();  
list.add("Java");
```

```
list.add("Python");  
System.out.println(list.poll());
```

- (a) Java
- (b) Python
- (c) true
- (d) Compilation error

Answer: a) Java

Explanation: The poll() method of LinkedList retrieves and removes the first element in a LinkedList. In this case, it retrieves and removes 'Java'.

Q#12. Which one of the following collection classes can be used to store unique & sorted objects?

- (a) HashSet
- (b) TreeSet
- (c) LinkedHashMap
- (d) ArrayList

Answer: b) TreeSet

Explanation: The TreeSet class provides a way to store unique elements in a sorted order. It can order the elements using ascending order. TreeSet does not preserve the insertion order of elements but elements are sorted by keys.

Q#13. Apart from providing methods for adding, removing, and examining elements in a specific order, which of the following interfaces provides methods like peek() and poll()?

- (a) Set
- (b) List

- (c) Queue
- (d) Map

Answer: c) Queue

Explanation: The Queue interface provides methods for adding, removing, and examining elements in a specific order, following the FIFO (First-In-First-Out) principle. It also includes additional queue-specific operations like peek() and poll().

Q#14. Which method is used to add an element to the end of an ArrayList?

- (a) addLast()
- (b) addEnd()
- (c) add()
- (d) append()

Answer: c) add()

Explanation: The add() method itself is used to add an element to the end of an ArrayList.

Q#15. Which collection class allows duplicate elements and maintains their insertion order?

- (a) HashSet
- (b) TreeSet
- (c) LinkedHashSet
- (d) HashMap

Answer: c) LinkedHashSet

Explanation: The LinkedHashSet class allows duplicate elements and maintains their insertion order. It combines the features of a HashSet and a LinkedList.

Q#16. Which method is used to check if a specific element is present in a Set?

- (a) `containsElement()`
- (b) `containsKey()`
- (c) `containsValue()`
- (d) `contains()`

Answer: d) `contains()`

Explanation: The `contains()` method is used to check if a specific element is present in a Set. If the element is found, It returns true, otherwise false.

Q#17. Which collection class stores objects in the form of key-value pairs and maintains the order in which they were inserted?

- (a) `Hashtable`
- (b) `LinkedHashMap`
- (c) `TreeMap`
- (d) `HashSet`

Answer: b) `LinkedHashMap`

Explanation: The `LinkedHashMap` class provides a way to store key-value pairs. It maintains the insertion order of the elements.

Q#18. Which collection class is synchronized and thread-safe?

- (a) `ArrayList`
- (b) `HashSet`
- (c) `ConcurrentHashMap`
- (d) `LinkedList`

Answer: c) `ConcurrentHashMap`

Explanation: The `ConcurrentHashMap` class is synchronized and thread-safe, allowing multiple threads to access it simultaneously without data discrepancies.

Q#19. Which collection class in Java provides a fixed-size collection that does not allow adding or removing elements?

- (a) ArrayList
- (b) LinkedList
- (c) HashSet
- (d) Arrays

Answer: d) Arrays Explanation: The Arrays class in Java provides utility methods for working with fixed-size arrays, which do not allow adding or removing elements once initialized.

Q#20. Which method is used to retrieve the first element from a Queue without removing it?

- (a) peek()
- (b) getFirst()
- (c) element()
- (d) front()

Answer: a) peek()

Explanation: The peek() method is used to retrieve the first element from a Queue without removing it. If the Queue is empty, it returns null.

Q#21. Which interface should be implemented to create a custom Comparator for sorting objects in a TreeSet?

- (a) Comparable
- (b) Sortable
- (c) Comparator
- (d) ComparatorSet

Answer: c) Comparator

Explanation: To create a custom Comparator for sorting objects in a TreeSet, the Comparator interface should be implemented.

Q#22. What is the output of the following code?

```
LinkedHashSet<Integer> set = new LinkedHashSet<>();  
set.add(10);  
set.add(20);  
set.add(30);  
System.out.println(set);
```

- (a) 10 20 30
- (b) [30, 20, 10]
- (c) [10, 20, 30]
- (d) Compilation error

Answer: c) [10, 20, 30]

Explanation: The LinkedHashSet maintains the insertion order of elements. It will output the elements in the order they were added.

Q#23. What is the output of the following code?

```
ArrayList<String> list = new ArrayList<>();  
list.add("Java");  
list.add("Python");  
System.out.println(list.contains("Python"));
```

- (a) true
- (b) false
- (c) Python
- (d) Compilation error

Answer: a) true

Explanation: The contains() method is used to check if a specific element exists in the ArrayList. In this case, it returns true as "Python" is present.

Q#24. Which method is used to obtain a synchronized (thread-safe) version of a List collection?

- (a) synchronized()
- (b) synchronizedList()
- (c) synchronizedCollection()
- (d) synchronize()

Answer: b) synchronizedList()

Explanation: The synchronizedList() method is used to obtain a synchronized (thread-safe) version of a List collection, such as ArrayList.

Q#25. What is the output of the following code?

```
TreeSet<Integer> set = new TreeSet<>();  
set.add(24);  
set.add(5);  
set.add(50);  
System.out.println(set.first());
```

- (a) 10
- (b) 5
- (c) 20
- (d) Compilation error

Answer: b) 5

Explanation: The first() method of the TreeSet is used to retrieve the first (lowest) element. In this case, it retrieves the smallest element, which is 5.

Q#26. Which method is used to retrieve the last element without removing it from a LinkedList?

- (a) `getLast()`
- (b) `lastElement()`
- (c) `getLastElement()`
- (d) `peekLast()`

Answer: a) `getLast()`

Explanation: The `getLast()` method is used to retrieve the last element from a `LinkedList`. It returns the last element without removing it from the list.

Q#27. What is the output of the following code snippet?

```
List<String> list = new ArrayList<>();  
list.add("Java");  
list.add("Python");  
list.add("Scala");  
System.out.println(list.subList(1, 3));
```

- (a) `[Java, Python]`
- (b) `[Python, Scala]`
- (c) `[Python]`
- (d) `[Scala]`

Answer: b) `[Python, Scala]`

Explanation: The `subList(1, 3)` method returns a fraction of the list from index 1 (inclusive) to index 3 (exclusive), so the output is `[Python, Scala]`.

Q#28. What is the output of the following code?


```
LinkedHashSet<String> set = new LinkedHashSet<>();  
set.add("Java");  
set.add("Python");  
System.out.println(set.remove("Scala"));
```

- (a) true
- (b) false
- (c) Java
- (d) Compilation error

Answer: b) false

Explanation: The remove() method is used to remove a specific element from a LinkedHashSet. In this case, "Scala" is not present, so it returns false.

Q#29. Which method is used to retrieve the key associated with the maximum value in a TreeMap?

- (a) getMaxKey()
- (b) firstKey()
- (c) lastKey()
- (d) getMaxValue()

Answer: c) lastKey()

Explanation: The lastKey() method is used to retrieve the key associated with the largest value in a TreeMap. Since the TreeMap is ordered by keys, the largest key is the last key.

Q#30. What is the output of the following code?

```
HashSet<Integer> set1 = new HashSet<>();  
set1.add(1);  
set1.add(2);
```

```
set1.add(3);

HashSet<Integer> set2 = new HashSet<>();
set2.add(2);
set2.add(3);
set1.add(4);

set1.retainAll(set2);
System.out.println(set1);
```

- (a) [1, 4]
- (b) [2, 3]
- (c) [1, 2, 3]
- (d) [1, 2, 3, 4]

Answer: b) [2,3]

Explanation: The retainAll() method is used to retain only the elements that exist in both sets. In this case, the intersection of set1 and set2 is [2,3].

Q#31. What is the output of the following code?

```
HashMap<String, Integer> map = new HashMap<>();
map.put("Java", 10);
map.put("Python", 20);
System.out.println(map.keySet());
```

- (a) ["Java", "Python"]
- (b) [10, 20]
- (c) "Java", "Python"
- (d) [Java, Python]

Answer: d) [Java, Python]

Explanation: The keySet() method is used to retrieve a set of all the keys in a HashMap. In this case, it returns [Java, Python].

Q#32. Which method is used to check if a LinkedList is empty?

- (a) isEmpty()
- (b) checkEmpty()
- (c) hasElements()
- (d) hasData()

Answer: a) isEmpty()

Explanation: The isEmpty() method is used to check if a LinkedList is empty. It returns true if the LinkedList contains no elements.

Q#33. What is the output of the following code?

```
LinkedList<String> list = new LinkedList<>();
list.add("Java");
list.add("Python");
list.add("Scala");
System.out.println(list.removeLast());
```

- (a) Java
- (b) Python
- (c) Scala
- (d) Compilation error

Answer: c) Scala

Explanation: The removeLast() method is used to retrieve and remove the last element in a LinkedList. In this case, it retrieves and removes "Scala".

Q#34. What happens when an element is added to a CopyOnWriteArrayList while multiple threads are iterating over it?

- (a) The newly added element is immediately visible to all iterators.
- (b) Iterators throw a ConcurrentModificationException.
- (c) Iterators continue without considering the newly added element.
- (d) The iteration blocks until the addition is complete.

Answer: c) Iterators continue without considering the newly added element.

Explanation: CopyOnWriteArrayList iterators operate on a copy of the underlying array, so they do not see modifications made after the iterator was created.

Q#35. What is the main difference between HashSet and TreeSet in Java?

- (a) HashSet allows duplicate elements, while TreeSet does not.
- (b) HashSet maintains insertion order, while TreeSet does not.
- (c) HashSet is unsorted, while TreeSet is sorted.
- (d) TreeSet allows null values, while HashSet does not.

Answer: c) HashSet is unsorted, while TreeSet is sorted.

Explanation: HashSet does not maintain any specific order, while TreeSet sorts elements in a specific order.

Q#36. Which collection class in Java does not allow null values?

- (a) HashSet
- (b) ArrayList
- (c) HashMap
- (d) TreeSet

Answer: d) TreeSet

Explanation: TreeSet does not allow null values because it uses the natural ordering of elements or a custom comparator, which cannot compare null.

Q#37. What is the difference between ArrayList and LinkedList in terms of inserting an element in the middle of the collection?

- (a) ArrayList is faster than LinkedList.
- (b) LinkedList is faster than ArrayList.
- (c) Both ArrayList and LinkedList have the same performance.
- (d) It depends on the size of the collection.

Answer: b) LinkedList is faster than ArrayList.

Explanation: LinkedList performs better than ArrayList when inserting elements in the middle because it does not require shifting elements.

Q#38. Which collection class in Java is best suited for frequent search operations on a large collection?

- (a) HashSet
- (b) ArrayList
- (c) LinkedList
- (d) TreeMap

Answer: d) TreeMap

Explanation: TreeMap uses a balanced tree structure, providing efficient search operations (logarithmic time complexity) on large collections.

Q#39. What is the output of the following code snippet?

```
List<Integer> list = new ArrayList<>();  
list.add(4);  
list.add(2);  
list.add(3);
```

```
list.add(2);  
System.out.println(list.lastIndexOf(2));
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4

Answer: c) 3

Explanation: The `lastIndexOf(2)` method returns the index of the last occurrence of the specified element in the list. In this case, the last occurrence of 2 is at index 3, so the output is 3.

Q#40. Which statement(s) is/are true about `CopyOnWriteArrayList` in Java?

- (a) `CopyOnWriteArrayList` class implements `List` and `RandomAccess` interfaces.
- (b) `CopyOnWriteArrayList` is a concurrent replacement for a synchronized `List`.
- (c) It allows duplicate elements and heterogeneous Objects.
- (d) It throws `ConcurrentModificationException`.

Answer: (a), (b), (c)

Explanation: `CopyOnWriteArrayList` is one of the solution for `ConcurrentModificationException`. Hence it can't throw `ConcurrentModificationException`. All other statements are true for `CopyOnWriteArrayList`.

Q#41. What is a key feature of a Java Map object?

- a) It allows any value as a key
- b) It maintains insertion order of elements
- c) It automatically removes duplicate keys
- d) It stores key-value pairs

Answer: d) It stores key-value pairs

Explanation: The primary function of a Java Map is to store key-value pairs, allowing for efficient retrieval, insertion, and updating of elements based on their keys. Although Java Maps can use various types of objects as keys, not all values are valid keys. For example, keys must correctly implement hashCode() and equals() methods. Insertion order behavior is specific to certain implementations like LinkedHashMap, but it is not a general feature of all Java Map implementations. Java Maps do not allow duplicate keys. If a key that already exists in the map is added, the new value will overwrite the old value associated with that key.

codewitharrays.in 8007592194



<https://www.youtube.com/@codewitharrays>



<https://www.instagram.com/codewitharrays/>



<https://t.me/codewitharrays> Group Link: <https://t.me/ccee2025notes>



[+91 8007592194](tel:+918007592194) [+91 9284926333](tel:+919284926333)



codewitharrays@gmail.com



<https://codewitharrays.in/project>