

Top 80 Java Collections Interview Questions & Answers

Here are Java Collections Interview Questions for fresher as well as experienced candidates to get their dream job.

1) What is framework in Java?

A framework is a popular and readymade architecture that contains a set of classes and interfaces.

2) What is the Collection framework in Java?

Collection Framework is a grouping of classes and interfaces that is used to store and manage the objects. It provides various classes like Vector, ArrayList, HashSet, Stack, etc. Java Collection framework can also be used for interfaces like Queue, Set, List, etc.

3) Explain Collections Class

`java.util.Collections` is a class consists of static methods that operate on collections. It contains polymorphic algorithms to operate on collections, "wrappers". This class contains methods for algorithms, like binary sorting, search, shuffling, etc.

4) What is the hashCode()?

The `hashCode()` is a method that returns an integer hash code.



([./images/2/062520_0728_Top80JavaCo1.png](#)).

5) Distinguish between ArrayList and Vector in the Java collection framework.



What is JVM (Java Virtual Machine) with Architecture JAVA Programming Tutorial



Learn Java Programming with Beginners Tutorial

ArrayList

ArrayList is cannot be synchronized.

It is not a legacy class.

It can increase its size by 50% of the size of the array.

ArrayList is not thread-safe.

Vector

Vector can be synchronized.

It is a legacy class.

It can increase its size by doubling the size of the array.

Vector is a thread-safe.

6) What is ArrayList in Java?

ArrayList is a data structure that can be stretched to accommodate additional elements within itself and shrink back to a smaller size when elements are removed. It is a very important data structure useful in handling the dynamic behavior of elements.

7) Differentiate between Iterator and ListIterator

The difference between Iterator and ListIterator is:

Iterator

The Iterator can traverse the array elements in the forward direction.

It can be used in Queue, List, and Set.

It can perform only remove operation.

ListIterator

ListIterator can traverse the array elements in backward as well as forward directions.

It can be used in List.

It can perform add, remove, and set operation while traversing the collection.

8) What is the difference between Iterator and Enumeration?

The difference between Iterator and Enumeration

Iterator

Enumeration



The Iterator can traverse both legacies as well as non-legacy elements.	Enumeration can traverse only legacy elements.
The Iterator is fail-fast.	Enumeration is not fail-fast.
The Iterator is very slow compare to Enumeration.	Enumeration is fast compare to Iterator.
The Iterator can perform remove operation while traversing the collection.	The Enumeration can perform only traverse operation on the collection.

9) Define BlockingQueue

BlockingQueue is an interface used in Java that can extend the Queue. It provides concurrency in various queue operations like retrieval, insertion, deletion, etc.

The Queue waits to become non-empty at the time of retrieving any elements. BlockingQueue should not contain null elements. The implementation of this Queue is thread-safe.

The syntax of BlockingQueue is:

```
public interface BlockingQueue<E> extends Queue <E>
```

10) Explain override equals() method

The equals method is used to check the similarity between two objects. In case if the programmer wants to check an object based on the property, then it needs to be overridden.

11) What is the difference between Comparable and Comparator?

The difference between Comparable and Comparator is:

Comparable	Comparator
Comparable provides compareTo() method to sort elements in Java.	Comparator provides compare() method to sort elements in Java.
Comparable interface is present in java.lang package.	Comparator interface is present in java.util package.
The logic of sorting must be in the same class whose object you are going to sort.	The logic of sorting should be in a separate class to write different sorting based on different attributes of objects.

The class whose objects you want to sort must implement the comparable interface.	Class, whose objects you want to sort, do not need to implement a comparator interface.
It provides single sorting sequences.	It provides multiple sorting sequences.
This method can sort the data according to the natural sorting order.	This method sorts the data according to the customized sorting order.
It affects the original class. i.e., the actual class is altered.	It doesn't affect the original class, i.e., the actual class is not altered.
Implemented frequently in the API by Calendar, Wrapper classes, Date, and String.	It is implemented to sort instances of third-party classes.
All wrapper classes and String class implement the comparable interface.	The only implemented classes of Comparator are Collator and RuleBasedColator.

12) Explain equals() with example

Equals() verifies whether the number object is equal to the object, which is passed as an argument or not.

The syntax of the equals() method is:

```
public boolean equals(Object o)
```

This method takes two parameters 1) any object, 2) return value. It returns true if the passed argument is not null and is an object of a similar type having the same numeric value.

Example:

```
import java.lang.Integer;
public class Test {
    public static void main(String args[]) {
        Integer p = 5;
        Integer q = 20;
        Integer r = 5;
        Short s = 5;
        System.out.println(p.equals(q));
        System.out.println(p.equals(r));
        System.out.println(p.equals(s));
    }
}
```

13) List out benefits of generic collection

The benefits of using the generic collection are:

- If the programmers are using generic class, they don't require typecasting.
- It is type-safe and can be checked at the time of compilation.
- It provides the stability of the code by detecting bug at the compilation time.

14) Explain the method to convert ArrayList to Array and Array to ArrayList

Programmers can convert an Array to ArrayList using `asList()` method of Arrays class. It is a static method of Arrays class that accept the List object. The syntax of `asList()` method is:

```
Arrays.asList(item)
```

Java programmers can convert ArrayList to the List object using syntax:

```
List_object.toArray(new String[List_object.size()])
```

15) Give example of ArrayList

The Example of reverse ArrayList is:

```
import java.util.ArrayList;
class Test_ArrayList {
    public static void main(String[] args) {
        //Creating a generic ArrayList
        ArrayList<String> arlTest = new ArrayList<String>();
        //Size of arrayList
        System.out.println("Size of ArrayList at creation: " + arlTest.size());
        //Lets add some elements to it
        arlTest.add("D");
        arlTest.add("U");
        arlTest.add("K");
        arlTest.add("E");

        //Recheck the size after adding elements
        System.out.println("Size of ArrayList after adding elements: " + arlTest.size());

        //Display all contents of ArrayList
        System.out.println("List of all elements: " + arlTest);

        //Remove some elements from the list
        arlTest.remove("D");
        System.out.println("See contents after removing one element: " + arlTest);

        //Remove element by index
        arlTest.remove(2);
        System.out.println("See contents after removing element by index: " + arlTest);
        //Check size after removing elements
        System.out.println("Size of arrayList after removing elements: " + arlTest.size());
        System.out.println("List of all elements after removing elements: " + arlTest);
        //Check if the list contains "K"
        System.out.println(arlTest.contains("K"));
    }
}
```

16) Give example to sort an array in descending order

The example of sort an array in decending order is:

```
package com.guru99;

public class SelectionSortAlgo {

    public static void main(String a[])
    {
        int[] myArray = {860,8,200,9};

        System.out.println("-----Before Sort-----");

        printArray(myArray);
        selection(myArray); //sorting array using selection sort
        System.out.println("-----After Sort-----");
        printArray(myArray);
    }

    public static void selection(int[] array)
    {
        for (int i = 0; i < array.length - 1; i++)
        {   System.out.println("Sort Pass Number "+(i+1));
            int index = i;
            for (int j = i + 1; j < array.length; j++)
            {
                System.out.println("Comparing "+ array[index] +
" and " + array[j]);
                if (array[j] < array[index]){
                    System.out.println(array[index] + " is greater
than " + array[j] );
                    index = j;
                }
            }
            int smallerNumber = array[index];
            array[index] = array[i];
            array[i] = smallerNumber;
            System.out.println("Swapping Elements: New Array Afte
r Swap");
            printArray(array);
        }
    }

    static void printArray(int[] array){

        for(int i=0; i < array.length; i++)
        {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }
}
```

17) Explain the basic interfaces of the Java collections framework

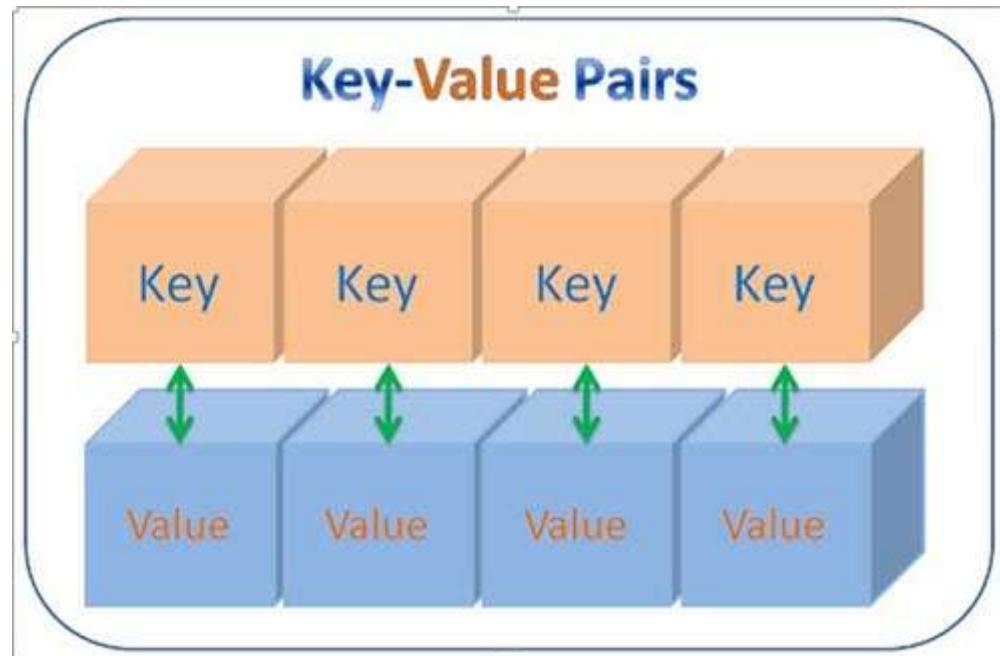
Java collection framework is a root of the collection hierarchy. It represents a group of objects as its elements. The Java programming language does not provide a direct implementation of such interface.

- **Set:** Set is a collection having no duplicate elements. It uses hashtable for storing elements.
- **List:** List is an ordered collection that can contain duplicate elements. It enables developers to access any elements from its inbox. The list is like an array having a dynamic length.
- **MAP:** It is an object which maps keys to values. It cannot contain duplicate keys. Each key can be mapped to at least one value.

18) What are the features of Java Hashmap?

Features of Java Hashmap are:

- The values can be stored in a map by forming a key-value pair. The value can be retrieved using the key by passing it to the correct method.
- If no element exists in the Map, it will throw a 'NoSuchElementException'.
- HashMap stores only object references. That is why it is impossible to use primitive data types like double or int. Use wrapper class (like Integer or Double) instead.



(/images/2/062520_0728_Top80JavaCo2.jpg).

19) What is a Stack?

A stack is a special area of computer's memory that stores temporary variables created by a function. In stack, variables are declared, stored, and initialized during runtime.

20) What is linked list?

A linked list is a data structure that can store a collection of items. In other words, linked lists can be utilized to store several objects of the same type. Each unit or element of the list is referred as a node. A node in the Linked list has its data and the address of the next node. It is like a chain. Linked Lists are used to create graphs and trees.

21) Give example of ArrayList

The example of ArrayList is:

```

import java.util.ArrayList;
class Test_ArrayList {
    public static void main(String[] args) {
        //Creating a generic ArrayList
        ArrayList<String> arlTest = new ArrayList<String>();
        //Size of arrayList
        System.out.println("Size of ArrayList at creation: " + arlTest.size());
        //Lets add some elements to it
        arlTest.add("D");
        arlTest.add("U");
        arlTest.add("K");
        arlTest.add("E");

        //Recheck the size after adding elements
        System.out.println("Size of ArrayList after adding elements: " + arlTest.size());

        //Display all contents of ArrayList
        System.out.println("List of all elements: " + arlTest);

        //Remove some elements from the list
        arlTest.remove("D");
        System.out.println("See contents after removing one element: " + arlTest);

        //Remove element by index
        arlTest.remove(2);
        System.out.println("See contents after removing element by index: " + arlTest);

        //Check size after removing elements
        System.out.println("Size of arrayList after removing elements: " + arlTest.size());
        System.out.println("List of all elements after removing elements: " + arlTest);
        //Check if the list contains "K"
        System.out.println(arlTest.contains("K"));
    }
}

```

22) Explain linked list supported by Java

Two types of linked list supported by Java are:

- **Singly Linked list:** Singly Linked list is a type of data structure. In a singly linked list, each node in the list stores the contents of the node and a reference or pointer to the next node in the list. It does not store any reference or pointer to the previous node.

- **Doubly linked lists:** Doubly linked lists are a special type of linked list wherein traversal across the data elements can be done in both directions. This is made possible by having two links in every node, one that links to the next node and another one that connects to the previous node.

23) Explain the methods provided by the Queue interface?

Methods of Java Queue interface are:



Method	Description
boolean add(object)	Inserts specified element into the Queue. It returns true in case it a success.
boolean offer(object)	This method is used to insert the element into the Queue.
Object remove()	It retrieves and removes the queue head.
Object poll()	(): It retrieves and removes queue head or return null in case if it is empty.
Object poll()	It retrieves and removes queue head or return null in case if it is empty.
Object element()	Retrieves the data from the Queue, but does not remove its head.
Object peek()	Retrieves the data from the Queue but does not remove its head, or in case, if the Queue is the Queue is empty, it will retrieve null.

24) Mention the methods provided by Stack class

Important methods provided by Stack class are:

- **push():** Push item into the stack.
- **empty ():** This method finds that whether the stack is empty or not.
- **pop ():** This Java collection framework method removes the object from the stack.
- **search ():** This method searches items in the stack.
- **peek ():** This Java method looks at the stack object without removing it.

25) Define emptySet() in the Java collections framework

Method emptySet() that returns the empty immutable set whenever programmers try to remove null elements. The set which is returned by emptySet() is serializable. The syntax of this method is:

```
public static final <T> Set<T> emptySet()
```

26) Differentiate between Collection and Collections

The difference between Collection and Collections are:

Collection	Collections
The collection is an interface.	Collections is a class.
It represents a group of objects as a single entity.	It defines various utility methods for collection objects.
The collection is the root interface of the Java Collection framework.	Collections is a general utility class.
This interface is used to derive the collection data structures.	This class contains static methods to manipulate data structure.

27) Define LinkedHashSet in the Java Collection framework?

LinkedHashSet is a subclass of the class called HashSet and implements the set interface. It is a well-ordered version of HashSet that maintains a doubly-linked List across its all elements.

28) What is the difference between failfast and failsafe?

Failfast	Failsafe
----------	----------

It does not allow collection modification while iterating.	It allows collection modification while iterating.
It can throw ConcurrentModificationException	It can't throw any exception.
It uses the original collection to traverse the elements.	It uses an original collection copy to traverse the elements.
There is no requirement of extra memory.	There is a requirement of extra memory.

29) List collection views of a map interface

Collection views of map interface are: 1) key set view, 2) value set view, and 3) entry set view.

30) What are the benefits of the Collection Framework in Java?

The benefits of Collection Framework in Java are:

- Java collection framework offers highly efficient and effective data structures that enhance the accuracy and speed of the program.
- The program developed with the Java collection framework is easy to maintain.
- A developer can mix classes with other types that result in increasing the reusability of code.
- The Java collection framework enables programmers to modify the primitive collection types the way they like.

31) What is a good way to sort the Collection objects in Java?

A good way to sort Java collection objects is using Comparable and Comparator interfaces. A developer can use Collections.sort(), the elements are sorted based on the order mention in compareTo().

When a developer uses Collections.sort(Comparator), it sorts the objects depend on compare() of the Comparator interface.

32) Explain Vector in Java

The vector is the same as an array. It has components that can be accessed using an index value. Vectors can contain a legacy method that is not part of the collection framework.

33) What is the difference between Set and Map?

Set	Map
Set belongs to package java.util.	The map belongs package java.util.
It can extend the collection interface.	It does not extend the collection interface.
It does not allow duplicate values.	It allows duplicate values.
Set can sort only one null value.	The map can sort multiple null values.

34) Define dictionary class

The Dictionary class is a Java class that has a capability to store key-value pairs.

35) Define EnumSet

java.util.EnumSet is Set implementation that can be used with enum types. EnumSet having all elements must come from one enum type specified explicitly or implicitly. It is not synchronized, and also null keys are not allowed. EnumSet provides methods like EnumSet.of(E first, E... rest), complementOf(EnumSet s), and copyOf(Collection c).

36) What are the two ways to remove duplicates from ArrayList?

Two ways to remove duplicates from ArrayList are:

- **HashSet:** Developer can use HashSet to remove the duplicate element from the ArrayList. The drawback is it cannot preserve the insertion order.
- **LinkedHashSet:** Developers can also maintain the order of insertion by using LinkedHashSet instead of HashSet.

37) What is IdentityHashMap?

IdentityHashMap is a class that implements Serializable, Clonable interfaces, Map, and extends AbstractMap class. It is designed for the case wherein there is a need of reference-equality semantics.

38) What is WeakHashMap?

WeakHashMap is an implementation of the Java Map. It is used to store weak references to its keys. Sorting using this Map allows a key-value pair is collected as garbage. Its key is not referenced outside WeakHashMap.

39) What are the methods to make collection thread-safe?

The methods to make collection thread safe are:

- Collections.synchronizedList(list);
- Collections.synchronizedMap(map);
- Collections.synchronizedSet(set);

40) Explain UnsupportedOperationException

UnsupportedOperationException is an exception which is thrown on methods that are not supported by actual collection type.

For example, Developer is making a read-only list using "Collections.unmodifiableList(list)" and calling call(), add() or remove() method. It should clearly throw UnsupportedOperationException.

41) Name the collection classes that gives random element access to its elements

Collection classes that give random element access to its elements are: 1) ArrayList, 2) HashMap, 3) TreeMap, and 4) Hashtable.

42) Explain the difference between Queue and Deque.

Queue	Deque
It is called a single-ended Queue	It is called a double-ended Queue
Elements in the Queue are added or removed from one end	Elements in the Queue are added from either end can be added and removed from the both end
It is less versatile.	It is more versatile.

43) Mention the implementing List and Set interface

Class implementing List interface: 1) ArrayList, 2) Vector, and 3) LinkedList.

Class implementing Set interface: 1) HashSet, and 2) TreeSet.

44) Explain the design pattern followed by Iterator



The iterator follows the detail of the iterator design pattern. It provides developer to navigate through the objects collections using a common interface without knowing its implementation.

45) What is the peek() of the Queue interface?

Peek () is a method of queue interface. It retrieves all the elements but does not remove the queue head. In case if the Queue is empty, then this method will return null.

46) What is CopyOnWriteArrayList?

CopyOnWriteArrayList is a variant of ArrayList in which operations like add and set are implemented by creating a copy of the array. It is a thread-safe, and thereby it does not throw ConcurrentModificationException. This ArrayLists permits all the elements, including null.

47) Differentiate between ArrayList and LinkedList

The difference between ArrayList and LinkedList is:

ArrayList	LinkedList
It uses a dynamic array.	It uses a doubly-linked list.
ArrayList is not preferable for manipulation.	LinkedList is preferable for manipulation.
ArrayList provides random access.	LinkedList does not provide random access.
ArrayList stores only objects hence it takes less overhead of memory	LinkedList stores object as well as address object; hence, it takes more overhead of memory.

48) Explain the methods of iterator interface

Methods of iterator interface are:

Method	Description
public boolean hasNext()	It returns true in the iterator has elements; otherwise, it returns false.
public Object next()	This method returns the element and moves the pointer to the next value.
public void remove()	This Java method can remove the last elements returned by the iterator. Public void remove() is less used.



49) What are the methods of the HashSet class?

Methods of HashSet class are:

Methods	Description
boolean add(Object o)	This method adds the mention element to this set if it is not already present.
boolean contains(Object o):	It returns true if the set contains the specified element.
void clear():	This method removes set elements.
boolean isEmpty():	It returns true in the case, the set has no elements.
boolean remove(Object o):	It remove the specified element from the set.
object clone():	This method returns a copy of the HashSet instance: the elements themselves are not cloned.
iterator iterator()	It returns an iterator over the elements in this set.
int size():	It returns the number of elements available in the set.

50) What are the methods of Java TreeSet class?

The methods of Java TreeSet class are:

Methods	Descriptions
boolean addAll(Collection c)	Add all the elements in the specified collection to this set.
boolean contains(Object o)	Returns true if the set contains the mention element.
boolean isEmpty()	This Java method returns true if this set contains no elements.
boolean remove(Object o)	Remove the specified element from the set.
void add(Object o)	It adds the specified element to the set.

void clear()

This Java method removes all the elements from the set.

51) Explain Linked HashSet

Java LinkedHashSet class is a Linked list and Hash table implementation of the Set interface. It contains unique elements same as a HashSet. Linked HashSet in Java also provides optional set operations that can maintain the order of insertion.

52) What are the important methods used in a linked list?

The important methods used in the linked list are:

Method	Description
boolean add(Object o)	It is used to append the specified element to the end of the vector.
boolean contains(Object o)	It a method that returns true if this list contains the specified element.
void add (int index, Object element)	Inserts the element at the specified element in the vector.
void addFirst(Object o)	It is used to insert the given element at the beginning.
void addLast(Object o)	It is used to append the given element to the end.
Int size()	This method can be used to return the total number of elements in a list.
boolean remove(Object o)	It can remove the first occurrence of the specified element from this list.
int indexOf(Object element)	This Java method returns the index with the first occurrence of the mention element in this list, or -1.
int lastIndexOf(Object element)	It is a Java method that returns the index with the last occurrence of the specified element in this list, or -1.

53) List various classes available in sets

Various classes available in sets are: HashSet, TreeSetand, and LinkedHashSet.

54) List methods available in Java Queue interface

- boolean add(object)
- boolean offer(object)
- object remove()
- object poll()
- object element()
- object peek()

55) Differentiate between List and Set.

List	Set
An ordered collection of elements	An unordered collection of elements
Preserves the insertion order	Doesn't preserves the insertion order
Duplicate values are allowed	Duplicate values are not allowed
Any number of null values can be stored	Only one null values can be stored
ListIterator can be used to traverse the List in any direction	ListIterator cannot be used to traverse a Set
Contains a legacy class called vector	Doesn't contains any legacy class

56) Explain for each loop with example

For-Each Loop is another form of for loop used to traverse the array. It reduces the code significantly, and there is no use of the index or rather the counter in the loop.

Example of for each loop:

```
class UsingForEach {
    public static void main(String[] args) {
        String[] arrData = {"Alpha", "Beta", "Gamma", "Delta", "Sigma"};
        //The conventional approach of using the for loop
        System.out.println("Using conventional For Loop:");
        for(int i=0; i< arrData.length; i++){
            System.out.println(arrData[i]);
        }
        System.out.println("\nUsing Foreach loop:");
        //The optimized method of using the for loop - also called the foreach lo
op
        for (String strTemp : arrData){
            System.out.println(strTemp);
        }
    }
}
```

57) Explain diamond operator

Diamond operator enables the compiler to collect the type arguments of generic class. In Java SE, developer can substitute the parameterized constructor with an empty parameter sets (<>) known as diamond operator.

58) Explain randomaccess interface

RandomAccess interface is used by List implementations for the indication that they are supporting fast.

59) Name the collection classes that implement random access interface

Java.util package has classes that can implement random access interface are:
CopyOnWriteArrayList, Stack, ArrayList, and Vector.

60) How to join multiple ArrayLists?

The list provides a addall() method multiple ArrayList in Java.

For example, consider two lists 1) areaList and 2) secondAreaList. A developer can join them using addall() like:

```
areaList.addAll(secondAreaList);
```

61) Explain deque Interface

Java.util.Deque is Java, an interface that extends Queue interface. It gives support for the insertion and deletion of elements at both the end. This Queue is also called a double-ended queue.

62) Explain LinkedHashMap

LinkedHashMap is the implementation of the Map interface. It can also extends the HashMap class. Therefore, like HashMap, LinkedHashMap enables Java developers to allow one null key and more than one null value.

63) Explain methods to remove elements from ArrayList

The methods to remove elements from ArrayList are:

Method	Description
clear()	This method removes the elements from ArrayList.
remove(int index)	This method of ArrayList can remove the element at a particular position.
remove(Object o)	It can remove the first occurrence of the mention element from the ArrayList.
removeAll()	It can remove the list of elements that are in a particular collection.
removeIf(Predicate<? super E> filter)	This method removes elements that satisfy the mention of a predicate.

64) Explain map.entry In Map

Map.entry is a Java interface of java.util. It has a nested interface in Map. This interface must be qualified by the name of class or interface, which it is a member. Therefore it is qualified as a Map.Entry. It represents a key and value pair that can forms element of a Map.

This method returns a view of the collection. For example, consider cityMap as a map. The developer can use entrySet() to get the set view of map having an element Map.Entry. Programmer can also use getKey() and getValue() of the Map.Entry to get the pair of key and value of the map.

65) Which method is used to sort an array in ascending order?

Java collection framework method, Collections.sort() is used to sort an array in ascending order.

66) How to measure the performance of an ArrayList?

The performance of ArrayList can be measured by:

- **Adding an element:** Developer can add an element at the end of ArrayList using add(E e) method. It is O(1). In the worst scenario, it might go to O(n). This can happen if the developer adds more elements than the array capacity.
- **Retrieving an element:** - Developer can access the array index using get(int index). The performance, in this case, can be measured using ArrayList get() is O(1).
- **Removing an element:** In case, if the developers are removing element using the remove(int index), then the performance of ArrayList can be calculated using said remove(int index) operation is O(n - index) method.

67) Explain LinkedList class

LinkedList class in Java implements Deque and List using a doubly linked list. There is a private class node in a doubly-linked list which provides its structure. It also has an item variable for holding the value and reference to Node class. This can be used for connecting the next and previous nodes.

68) Give an example of Hashmap

The example of Hashmap is:

```
import java.util.HashMap;
import java.util.Map;
public class Sample_TestMaps{
    public static void main(String[] args){
        Map<String, String> objMap = new HashMap<String, String>();
        objMap.put("Name", "Suzuki");
        objMap.put("Power", "220");
        objMap.put("Type", "2-wheeler");
        objMap.put("Price", "85000");
        System.out.println("Elements of the Map:");
        System.out.println(objMap);
    }
}
```

69) How to iterate map?

The developer cannot directly iterate map, but, this interface has two methods that gives view set of map. These methods are:

- **Set<Map.Entry<K, V>>entrySet():** It is a method that returns a set having the entries mentioned in the map. These entries are generally objects, which have type Map. Entry.
- **Set<K>keySet():** This Java method returns a set that has the map key.

70) Explain Treemap in Java

TreeMap is a class that implements the Map interface LinkedHashMap and HashMap. It can also implement the NavigableMap interface and extends the AbstractMap class.

71) What is the difference between Hashmap and Hashtable?

Hashmap	Hashtable
It is not synchronized.	It is synchronized.
HashMap allows one key as a null value.	HashTable does not allow null values.
Iterator is used to traverse HashMap.	Either Iterator or Enumerator is used for traversing a HashTable.
It can be used for both HashTable, HashMap and is fail-fast.	It can be used with HashTable and is fail-safe.
HashMap performs faster than the HashTable.	Hashtable is not much faster as compared to HashMap.

72) Explain the internal working of HashSet in Java

HashSet in Java internally uses HashMap to store elements. It can also store unique values with no duplicate values.

In Java, HashSet developer can have add(E e) method that takes just the element to add as a parameter. It does not accept the key and value pair.

73) Explain Big-O notation with an example

The Big-O notation depicts the performance of an algorithm as the number of elements in ArrayList. A developer can use Big-O notation to choose the collection implementation. It is based on performance, time, and memory.

For example, ArrayList get(index i) is a method to perform a constant-time operation. It does not depend on the total number of elements available in the list. Therefore, the performance in Big-O notation is O(1).

74) Explain the best practices in Java Collection Framework

The best practices in Java Collection Framework are:

- Choose the correct type of collection depends on the need.
- Avoid rehashing or resizing by estimating the total number of elements to be stored in collection classes.
- Write a Java program in terms of interfaces. This will help the developer to change it's implementation effortlessly in the future.
- A developer can use Generics for type-safety.
- Use immutable classes given by the Java Development Kit. Avoid implementation of equals() and hashCode() for custom classes.
- A programmer should use the Collections utility class for algorithms or to get read-only, synchronized, or empty collections. This will enhance code reusability with low maintainability.

75) Explain various types of queues in Java

There are three types of queues in Java:

- **Priority queue:** It is a special type of Queue wherein elements are sorted as per their natural ordering or custom comparator.
- **Circular Queue:** It is a type of Queue in which user operations are performed based on the FIFO method. The last element is connected to the first position in order to make a circle.
- **Double-ended Queue:** A double-ended queue is an abstract data type that generalizes a queue. The elements in this queue can be added or removed from either head or tail.

76) What is the difference between stack and Queue?

Stack

Queue

The working principle of the stack is LIFO.

One end is used to perform the insertion or deletion of elements.

It uses one pointer.

It does not have any kind of variant.

It is easy to use.

Working principle of queue is FIFO.

One end is used to perform insertion, and another end is used for the deletion of elements.

It uses two pointers in a simple queue.

It has variants like priority queue, circular Queue, doubly ended Queue.

It is not easy to use.

77) What is the difference between array and stack?

The difference between array and stack is:

Array

It is a collection of elements that are identified by the index.

It has elements of data types which are same.

Elements can be removed or added into the array using random access operation.

Stack

It is a collection operation that serve as operations push and pop.

It has elements of data types which are different.

Elements can be removed or added into a stack using LIFO operation.

78) Define Iterator()

The Iterator() is an interface that provides methods to iterate Collection. Iterator can take the place of Enumeration in Java. It allows the caller to remove elements from the collection. The method provides a generic way for traversal using elements of the collection and implementing iterator design pattern.

79) What are the various ways to iterate over a list?

Java collection Framework programmer can iterate over a list in two ways: 1) Using iterator, and 2) using it for each loop.

80) What are the advantages of the stack?

The advantages of the stack are:

- It helps you to manage the data in a Last In First Out (LIFO) method, which is not possible with the Linked list and array.
- When a function is called, the local variables are stored in a stack, and it is automatically destroyed once returned.
- A stack is used when a variable is not used outside that function.
- It allows you to control how memory is allocated and deallocated.
- Stack automatically cleans up the object.
- Not easily corrupted
- Variables cannot be resized.

◀ [Prev \(/java-interview-questions-answers.html\)](/java-interview-questions-answers.html)

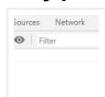
[Report a Bug](#)

[Next ➡ \(/java-design-patterns-interview-questions.html\)](#)

YOU MIGHT LIKE:

JAVASCRIPT

(/typescript-tutorial.html)



(/typescript-tutorial.html)

[TypeScript Tutorial: Types, Interfaces, Enums, Arrays, Example](#)

(/typescript-tutorial.html)

JAVA TUTORIALS

(/java-tutorial-pdf.html)



(/java-tutorial-pdf.html)

[JAVA Tutorial PDF: Basics PDF for Beginners \(Download Now\)](#)

(/java-tutorial-pdf.html)

JAVA TUTORIALS

(/buffered-reader-in-



java.html)

(/buffered-reader-in-
java.html)

[Java BufferedReader: How to Read File in Java with Example](#)

(/buffered-reader-in-
java.html)

JAVA TUTORIALS

(/java-user-defined-



exception.html)

(/java-user-defined-
exception.html)

[How to Create User Defined Exception in Java](#)

JAVA TUTORIALS

(/jasperreports-



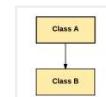
tutorial.html)

(/jasperreports-
tutorial.html)

[JasperReports Tutorial: What is, How to Install, Report Example](#)

JAVA TUTORIALS

(/java-class-



inheritance.html)

(/java-class-
inheritance.html)

[Inheritance in Java OOPs with Example](#)



Java Tutorials

74) Java Interview Q & A ([/java-interview-questions-answers.html](#))

50) Java Reflection API Tutorials ([/java-reflection-api.html](#))

75) Java Collections Interview Q & A ([/java-collections-interview-questions-answers.html](#)).

51) Java Tools for Developers ([/java-tools.html](#)).

52) BEST Java IDE ([/best-java-ide.html](#))

✉



 (<https://twitter.com/guru99com>),  (<https://www.linkedin.com/company/guru99/>).

<https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ>.

<https://forms.aweber.com/form/46/724807646.htm>.

About

[About Us \(/about-us.html\)](#),
[Advertise with Us \(/advertise-us.html\)](#),
[Write For Us \(/become-an-instructor.html\)](#),
[Contact Us \(/contact-us.html\)](#).

Career Suggestion

[SAP Career Suggestion Tool \(/best-sap-module.html\)](#),
[Software Testing as a Career \(/software-testing-career-complete-guide.html\)](#).

Interesting

[eBook \(/ebook-pdf.html\)](#),
[Blog \(/blog/\)](#),
[Quiz \(/tests.html\)](#),
[SAP eBook \(/sap-ebook-pdf.html\)](#).

Execute online

[Execute Java Online \(/try-java-editor.html\)](#),
[Execute Javascript \(/execute-javascript-online.html\)](#),
[Execute HTML \(/execute-html-online.html\)](#),
[Execute Python \(/execute-python-online.html\)](#).



are
ire
et
)

Top 20 Hibernate Interview Questions with Answers for Java Programmers

Hibernate is one of the most popular persistent frameworks in the Java world. Hibernate offers an object to relational (ORM) solution which frees Java developers from writing tedious, hard to read, and cluttered JDBC code converting SQL columns into Object properties. Apart from freeing Java developers from writing JDBC and database interaction code, Hibernate also offers the out-of-box solution on **caching**, proxying, and lazy loading which drastically improves the performance of your Java Web application. Given its importance in the Java Web application development field, Hibernate has become one of the most sought-after skills and goes hand in hand with Spring framework. That's why Spring and Hibernate interview questions are also very popular in Java interviews, particularly for web development positions.



Earlier I have shared some **Spring MVC interview questions** and due to popular demand, I am now sharing some **20 frequently asked Hibernate questions** from various Java interviews.

These questions are very useful for both the phone and the face-to-face round of interviews and both beginners and experienced Java developers up to 2 to 5 years can benefit from these questions. Since no list is complete without user involvement, so if you have seen any good Hibernate questions on your interviews, which are not on this list then feel free to share with us.

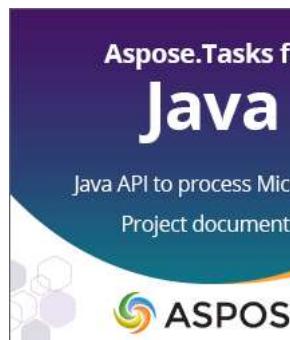
You can also share your own questions for which you are looking answers or you feel it's a good question to be asked on Java JEE programming interviews.

Btw, I expect that you have some Hibernate experience and knowledge before you go through this list of questions otherwise it would be quite back and forth. If you are a beginner to Hibernate, I suggest you first go through a course like **Spring and Hibernate for Beginners** which provides all the fundamental knowledge in quick time.

After that you can better understand the points interviewers are looking and these questions will also make sense to you.

Interview Q

- core java
- SQL int
- data str
- coding i
- java col
- java des
- question
- thread i
- hiberna
- j2ee int
- Spring I
- object o
- question



20 Hibernate Questions from Java JEE Interviews

Here is my selected list of 20 Hibernate based questions for Java developers. It contains questions from Hibernate fundamentals, one-to-one, and one-to-many mappings, caching, Hibernate vs JDBC comparison, pros and cons of Hibernate, known problems with Hibernate, and Performance improvement.

1. What is Hibernate?

Hibernate is an ORM (Object-relational Mapping) framework, which allows the developer to concentrate on business logic by taking care of the persistence of data by itself. Java developer can write code using an object and Hibernate can take care of creating those object from data loaded from the database and saving update back to the database.

2. What are the advantages of Hibernate over JDBC? (detailed answer)

Apart from Persistence i.e. saving and loading data from Database, Hibernate also provides the following benefits

- 1) Caching
- 2) Lazy Loading
- 3) Relationship management and provides code for mapping an object to the data
- 4) The developer is free from writing code to load/store data into the database.

3. Difference between get() vs load() method in Hibernate? (detailed answer)

This is one of the most frequently asked Hibernate interview questions, I have seen it several times. The key difference between the `get()` and `load()` method is that `load()` will throw an exception if an object with an id passed to them is not found, but `get()` will return null.

Another important difference is that `load` can return proxy without hitting the database unless required (when you access any attribute other than id) but `get()` always go to the database, so sometimes using `load()` can be faster than the `get()` method.

Use the `load()` method, if you know the object exists, and the `get()` method if you are not sure about the object's existence. See [Introduction To Hibernate](#) course on Pluralsight to learn more about how Hibernate works

Follow to receive notifications

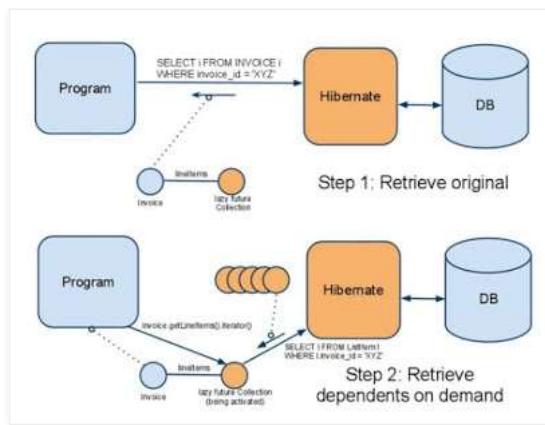
Email address

Recommend this article

- 10 Must Read Books for All Level
- 10 Frameworks You Should Learn
- 10 Books Every Developer Should Read in 2019
- 10 Open Source Java Frameworks
- 10 Programming Languages in 2018
- The Best Java Books for Developers
- 10 Java Questions Every Developer Should Know
- Top 10 Frameworks for Java
- 10 Books Every Java Developer Should Read
- 5 Great Java Books for Developers

Books and Courses

- Best Books for Java Beginners
- 5 Books Every Java Developer Should Read
- 2 books every Java developer should read
- 12 Advanced Java Books for Developers
- 5 Free Java Books for Developers
- 5 Books Every Java Developer Should Read
- Books Every Java Developer Should Read
- Top 10 Courses for Java Developers
- 10 Free Java Courses for Developers
- 5 Books Every Java Developer Should Read
- Books Every Java Developer Should Read



Btw, you would need a Pluralsight membership to access this course, which costs around \$29 monthly or \$299 annually. I have one and I also suggest all developers have that plan because Pluralsight is like NetFlix for Software developers.

It has more than 5000+ good quality courses on all the latest topics. Since we programmers have to learn new things every day, an investment of \$299 USD is not bad.

Btw, it also offers a **10-day free trial** without any obligation which allows you to watch 200 hours of content. You can watch this course for free by signing for that trial.

4. What is the N+1 SELECT problem in Hibernate? (detailed answer)

The N+1 SELECT problem is a result of lazy loading and load on demand fetching strategy. In this case, Hibernate ends up executing N+1 SQL queries to populate a collection of N elements.

For example, if you have a List of N Items where each Item has a dependency on a collection of Bid object. Now if you want to find the highest bid for each item then Hibernate will fire 1 query to load all items and N subsequent queries to load Bid for each item.

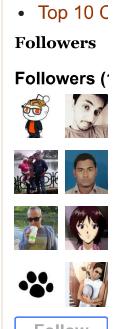
So in order to find the highest bid for each item your application ends up firing N+1 queries. It's one of the important Hibernate interview questions and I suggest reading chapter 13 of Java Persistence with Hibernate to understand this problem in more detail.

5. What are some strategies to solve the N+1 SELECT problem in Hibernate? (detailed answer)

This is the follow-up question of the previous Hibernate interview question. If you answer the last query correctly then you would be most likely asked this one.

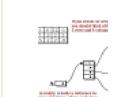
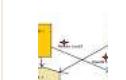
Here are some strategies to solve the N+1 problem:

- 1) pre-fetching in batches, will reduce the N+1 problem to $N/K + 1$ problem where K is the size of the batch
- 2) subselect fetching strategy
- 3) disabling lazy loading



How to Reverse engineer Java? [Simple Guide]

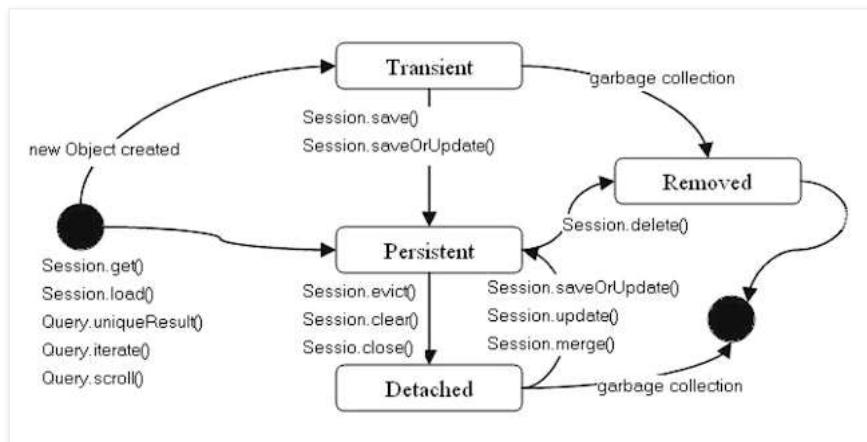
Hello guys,
reverse engineer
then you have
Earlier, I ha



5. What is the difference between save() and persist() method in Hibernate? (detailed answer)

The main difference between the `save()` and `persist()` method is that, `save` returns a `Serializable` object while the return type of the `persist()` method is void, so it doesn't return anything.

You can further join a comprehensive hibernate course like [Master Hibernate and JPA with Spring Boot in 100 Steps](#) to learn Hibernate fundamentals like this. I highly recommend that course. Also, here is a nice diagram that explains the state transition in Hibernate:



7. What is the requirement for a Java object to become a Hibernate entity object? (detailed answer)

It should not be final and must provide a default, no-argument constructor. See the detailed answer to learn more about the special requirement for a Java object to become a Hibernate Entity.

8. What are the different types of caches available in Hibernate? (detailed answer)

This is another common Hibernate interview question. Hibernate provides the out-of-box caching solution but there are many caches like first level cache, second-level cache, and query cache.

The first level cache is maintained at the Session level and cannot be disabled but the second-level cache is required to be configured with an external cache provider like EhCache.

9. What is the difference between the first and second level cache in Hibernate? (detailed answer)

This is again a follow-up of the previous Hibernate interview question. The first level cache is maintained at the Session level while the second level cache is maintained at a SessionFactory level and shared by all sessions. You can read [these books](#) to learn more about caching in Hibernate.

10. Does Hibernate Session interface thread-safe in Java? (detailed answer)

No, the Session object is not thread-safe in Hibernate and intended to be used with-in a single thread in the application.

are
ire

set
)

11. Does SessionFactory thread-safe in Hibernate? (detailed answer)

SessionFactory is both Immutable and thread-safe and it has just one single instance in the Hibernate application. It is used to create a Session object and it also provides caching by storing SQL queries stored by multiple sessions.

The second-level cache is maintained at the SessionFactory level. This can be a difficult and tricky question for less experienced Java developers who are not familiar with **thread-safety** and **Immutability**.

12. What is the difference between Session and SessionFactory in Hibernate? (detailed answer)

This is another popular Hibernate interview question, mostly at a telephonic round of interviews.



The main difference between Session and SessionFactory is that the former is a single-threaded, short-lived object while the latter is Immutable and shared by all Session.

It also lives until the Hibernate is running. Another difference between Session and SessionFactory is that the former provides first-level cache while SessionFactory provides the Second level cache.

13. What is the criterion query in hibernate? (detailed answer)

Criteria is a simplified API for retrieving entities by composing Criterion objects also known as Criterion query.

This is a very convenient approach for functionality like "search" screens where you can filter data on multiple conditions as shown in the following example:

```
List books = session.createCriteria(Book.class)
.add(Restrictions.like("name", "java%"))
.add(Restrictions.like("published_year", "2015"))
.addOrder(Order.asc("name"))
.list();
```

This can be a tough question if you are not using Hibernate on a daily basis, I have interviewed several Java developers who have used Hibernate but doesn't know about Criterion query or API. You can check out this list of [5 Hibernate and JPA courses for Java Programmers](#) to learn more about JPA and Hibernate.

14. What are other ORM frameworks? Any alternative of Hibernate?

This is a general question, sometimes asked to start the conversation and other times to finish the interview. EJB and TopLink from Oracle are two of the most popular alternative to the Hibernate framework.

15. What is the difference between the save() and saveOrUpdate() method of Hibernate? (detailed answer)

Though both `save()` and `saveOrUpdate()` method is used to store an object into Database, the key difference between them is that `save` can only INSERT records but `saveOrUpdate()` can either INSERT or UPDATE records.

are
ire
)
set



16. What is difference between getCurrentSession() and openSession() in Hibernate? (detailed answer)

An interesting Hibernate interview question as you might have used both `getCurrentSession()` and `openSession()` to obtain an instance of the Session object. I have left this question unanswered for you to answer or find an answer based on your experience.

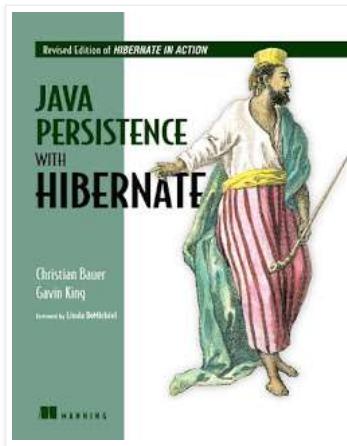
17. What is Hibernate Query Language (HQL)? (detailed answer)

Hibernate query language, HQL is an object-oriented extension to SQL. It allows you to query, store, update, and retrieve objects from a database without using SQL.

This question is also similar to the earlier question about Criterion query, Java developers who have not used Hibernate extensively will not know much about features like HQL and Criterion.

18. When do you use merge() and update() in Hibernate? (detailed answer)

This is one of the tricky Hibernate interview questions. You should use `update()` if you are sure that the Hibernate session does not contain an already persistent instance with the same id and use `merge()` if you want to merge your modifications at any time without considering the state of the session. See [Java Persistence with Hibernate](#) for more details.



19. The difference between sorted and ordered collections in Hibernate? (detailed answer)

The main difference between sorted and ordered collection is that sorted collection sort the data in JVM's heap memory using Java's collection framework sorting methods while the ordered collection is sorted using order by clause in the database itself.

A sorted collection is more suited for a small dataset but for a large dataset, it's better to use an ordered collection to avoid **OutOfMemoryError** in Java application.

20. How do you log SQL queries issued by the Hibernate framework in Java application?

You can use the `show_sql` property to log SQL queries issued by the Hibernate framework, Just add the following line in your Hibernate configuration file:

```
<property name="show_sql"> true </property>
```

21. What are the three states of a Hibernate Persistence object can be? (detailed answer)

The Hibernate persistent or entity object can live in the following three states:

- 1) transient
- 2) persistent
- 3) detached

Btw, If you are actively preparing for Java interviews then you can also check out **Java Programming Interview Exposed** book, it not only contains interview questions from Spring and Hibernate but also other important Java topics like core Java, data structure and algorithms, Servlet, JSP, JSF, and design patterns.

22. What is the difference between the transient, persistent, and detached state in Hibernate? (detailed answer)

New objects created in the Java program but not associated with any hibernate Session are said to be in the transient state.

On the other hand, an object which is associated with a Hibernate session is called a Persistent object. While an object which was earlier associated with the Hibernate session but currently it's not associate is known as a detached object.

You can call the `save()` or `persist()` method to store those objects into the database and bring them into the Persistent state. Similarly, you can re-attach a detached object to hibernate sessions by calling either the `update()` or `saveOrUpdate()` method.

See [Spring and Hibernate for Beginners](#) to learn more about the persistence object's lifecycle in Hibernate.

23. Which cache is used by Session Object in Hibernate? First level or second level cache? (detailed answer)

A Session object uses the first-level cache. As I told you before the second-level cache is used at the `SessionFactory` level. This is a good question to check if the Candidate has been working in hibernate or not. If he has not worked in Hibernate for a long time then he would get confused in this question.

That's all in this list of [Hibernate Interview questions for Java and JEE developers](#). In this article, we have covered a lot of frequently asked Hibernate questions for both beginners and experienced Java developers from all important topics of the Hibernate framework e.g. Hibernate fundamentals, caching, collection mapping, performance tuning, common issues, and Hibernate vs JDBC.

If you find that you lack knowledge in any particular area, I suggest go through the following resources to fill those gaps in your knowledge.

Further Learning

[Introduction To Hibernate](#)

[Spring and Hibernate for Beginners](#)

[Hibernate Interview Questions Preparation Course](#)

Other [Interview Questions articles](#) you may like

- [Top 20 Spring MVC Interview Questions](#)
- [Top 20 Spring and REST Interview Questions](#)
- [Top 20 Spring Boot Interview Questions with Answers](#)
- [5 Courses to Learn Spring Boot and Microservices](#)
- [5 Hibernate and JPA courses for Java Programmers](#)
- [5 Free Courses to learn Spring and Spring Boot](#)
- [Top 5 Spring and Hibernate Training courses](#)
- [130+ Java Interview Questions from last 5 years](#)
- [100+ Data Structure and Algorithm Questions](#)
- [10 Free Spring Boot Courses for Java Developers](#)
- [10 Best Spring Framework Courses for Java Programmers](#)
- [10 Advanced Spring Boot Courses for Developers](#)

Thanks for reading this article, if you like this article and interview question then please share with your friends and colleagues. If you have any questions or feedback then please

drop a comment.

P. S. - If you are just started with Java web development and looking for free courses to learn Spring, Spring MVC, and Spring Boot then you can also check out this list of **free Spring Courses from Udemy** for Java developers.

are
ire

set
)



DOWNLOAD YOUR FREE JAVA INTERVIEW GUIDE

Join my list for exclusive discounts, coupons, and early access to my books and courses.

And, download a free chapter from my best selling Grokking the Java interview book when you sign up

Email Address

[Download](#)

We respect your privacy. Unsubscribe at any time.

Posted by [javin paul](#) at [5:00 AM](#)



Labels: [Hibernate](#), [Hibernate interview Question](#), [Java Interview Question](#)

7 comments:

Anonymous February 10, 2016 at 2:03 AM

Thank you for sharing these questions. I was looking for latest Hibernate interview questions from long time. You can also include some practical questions like

which version of Hibernate you have used? e.g. Hibernate 3.0?

Do you use Annotation or config file for Mapping DB columns to object properties?

Have you used 2nd level cache in Hibernate? Which provider have you used?

What is benefit of Query caching?

I know you already have some of them included, but these are the way they were generally asked.

[Reply](#)



Anonymous January 10, 2017 at 5:30 PM

Below question(s) are also useful to judge candidate knowledge:

How we can integrate hibernate with spring?

What is @mappedBy annotation?

what all mapping hibernate supports?

Write a code persisting Employee object ?

[Reply](#)

Anonymous December 17, 2017 at 6:09 AM

describe isolation levels in database ?

what is phantom read ?

[Reply](#)

[Replies](#)



Unknown July 26, 2019 at 6:36 AM

Please add this question too...

what are the jars used in your project for hibernate.



javin paul July 26, 2019 at 8:16 PM

Great questions guys, keep them coming, I'll add them whenever I update the article, how about putting one or two lines about answers as well?

[Reply](#)



Pankaj Tiwari February 28, 2018 at 1:58 AM

nice tutorial for beginner and experience person.

[Reply](#)

[Replies](#)



javin paul July 26, 2019 at 8:15 PM

Thanks Pankaj.

[Reply](#)

are
ireset
)

Enter your comment...

 Comment as: Vegan (Google ▾) [Sign out](#)

[Publish](#) [Preview](#) [Notify me](#)

Feel free to comment, ask questions if you have any doubt.

[Newer Post](#) [Home](#) [Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Copyright by Soma Sharma 2012 to 2020. Powered by [Blogger](#).



Q1. What is the difference between an Inner Class and a Sub-Class?

Ans: An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class.

A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class.

Q2. What are the various access specifiers for Java classes?

Ans: In Java, access specifiers are the keywords used before a class name which defines the access scope. The types of access specifiers for classes are:

1. Public : Class,Method,Field is accessible from anywhere.
2. Protected: Method,Field can be accessed from the same class to which they belong or from the sub-classes, and from the class of same package, but not from outside.
3. Default: Method,Field, class can be accessed only from the same package and not from outside of its native package.
4. Private: Method,Field can be accessed from the same class to which they belong.

Q3. What's the purpose of Static methods and static variables?

Ans: When there is a requirement to share a method or a variable between multiple objects of a class instead of creating separate copies for each object, we use static keyword to make a method or variable shared for all objects.

Q4. What is data encapsulation and what's its significance?

Ans: Encapsulation is a concept in Object Oriented Programming for combining properties and methods in a single unit.

Encapsulation helps programmers to follow a modular approach for software development as each object has its own set of methods and variables and serves its functions independent of other objects. Encapsulation also serves data hiding purpose.

Q5. What is a singleton class? Give a practical example of its usage.

A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. Singleton class concept is useful for the situations when there is a need to limit the number of objects for a class.

The best example of singleton usage scenario is when there is a limit of having only one connection to a database due to some driver limitations or because of any licensing issues.

Q6. What are Loops in Java? What are three types of loops?

Ans: Looping is used in programming to execute a statement or a block of statement repeatedly. There are three types of loops in Java:

1) For Loops

For loops are used in java to execute statements repeatedly for a given number of times. For loops are used when number of times to execute the statements is known to programmer.

2) While Loops

While loop is used when certain statements need to be executed repeatedly until a condition is fulfilled. In while loops, condition is checked first before execution of statements.

3) Do While Loops

Do While Loop is same as While loop with only difference that condition is checked after execution of block of statements. Hence in case of do while loop, statements are executed at least once.

Q7: What is an infinite Loop? How infinite loop is declared?

Ans: An infinite loop runs without any condition and runs infinitely. An infinite loop can be broken by defining any breaking logic in the body of the statement blocks.

Infinite loop is declared as follows:

```
for (;;) {  
    // Statements to execute  
    // Add any loop breaking logic  
}
```

Q8. What is the difference between continue and break statement?

Ans: break and continue are two important keywords used in Loops. When a break keyword is used in a loop, loop is broken instantly while when continue keyword is used, current iteration is broken and loop continues with next iteration.

In below example, Loop is broken when counter reaches 4.

```
for (counter=0;counter<10;counter++)
```

```
system.out.println(counter);

if (counter==4) {

break;}

}
```

In the below example when counter reaches 4, loop jumps to next iteration and any statements after the continue keyword are skipped for current iteration.

```
for (counter=0;counter<10;counter++)
system.out.println(counter);

if (counter==4) {

continue;

}

system.out.println("This will not get printed when counter is 4");

}
```

Q9. What is the difference between double and float variables in Java?

Ans: In java, float takes 4 bytes in memory while Double takes 8 bytes in memory. Float is single precision floating point decimal number while Double is double precision decimal number.

Q10. What is Final Keyword in Java? Give an example.

Ans: In java, a constant is declared using the keyword Final. Value can be assigned only once and after assignment, value of a constant can't be changed.

In below example, a constant with the name const_val is declared and assigned avalue:

```
Private Final int const_val=100
```

When a method is declared as final,it can NOT be overridden by the subclasses.This method are faster than any other method,because they are resolved at complied time.

When a class is declares as final,it cannot be subclassed. Example String,Integer and other wrapper classes.

Q11. What is ternary operator? Give an example.

Ans: Ternary operator , also called conditional operator is used to decide which value to assign to a variable based on a Boolean value evaluation. It's denoted as ?

In the below example, if rank is 1, status is assigned a value of "Done" else "Pending".

```
public class conditionTest {  
    public static void main(String args[]) {  
        String status;  
        int rank = 3;  
        status = (rank == 1) ? "Done" : "Pending";  
        System.out.println(status);  
    }  
}
```

Q12: How can you generate random numbers in Java?

Ans:

- Using Math.random() you can generate random numbers in the range greater than or equal to 0.1 and less than 1.0
- Using Random class in package java.util

Q13. What is default switch case? Give example.

Ans: In a switch statement, default case is executed when no other switch condition matches. Default case is an optional case .

It can be declared only once all other switch cases have been coded.

In the below example, when score is not 1 or 2, default case is used.

```
public class switchExample {  
    int score=4;  
    public static void main(String args[]) {  
        switch (score) {  
            case 1:  
                system.out.println("Score is 1");  
                break;  
            case 2:  
                system.out.println("Score is 2");  
                break;  
            default:  
                system.out.println("Default Case");  
        }  
    }  
}
```

{
}**Q14. What's the base class in Java from which all classes are derived?**

Ans: java.lang.Object

Q15. Can main() method in Java can return any data?

Ans: In Java, main() method can't return any data and hence, it's always declared with a void return type.

Q16. What are Java Packages? What's the significance of packages?

Ans: In Java, package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once code has been packaged in Packages, it can be imported in other classes and used.

Q17. Can we declare a class as Abstract without having any abstract method?

Ans: Yes we can create an abstract class by using abstract keyword before class name even if it doesn't have any abstract method. However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error.

Q18. What's the difference between an Abstract Class and Interface in Java?

Ans: The primary difference between an abstract class and interface is that an interface can only possess declaration of public static methods with no concrete implementation while an abstract class can have members with any access specifiers (public, private etc) with or without concrete implementation.

Another key difference in the use of abstract classes and interfaces is that a class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn't require implementation of all the methods of its super class.

A class can implement multiple interfaces but it can extend only one abstract class.

Q19. What are the performance implications of Interfaces over abstract classes?

Ans: Interfaces are slower in performance as compared to abstract classes as extra indirections are required for interfaces. Another key factor for developers to take into consideration is that any class can extend only one abstract class while a class can implement many interfaces.

Use of interfaces also puts an extra burden on the developers as any time an interface is implemented in a class; developer is forced to implement each and every method of interface.

Q20. Does Importing a package imports its sub-packages as well in Java?

Ans: In java, when a package is imported, its sub-packages aren't imported and developer needs to import them separately if required.

For example, if a developer imports a package university.* , all classes in the package named university are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package (say department), developer has to import it explicitly as follows:

```
Import university.department.*
```

Q21. Can we declare the main method of our class as private?

Ans: In java, main method must be public static in order to run any application correctly. If main method is declared as private, developer won't get any compilation error however, it will not get executed and will give a runtime error.

Q22. How can we pass argument to a function by reference instead of pass by value?

Ans: In java, we can pass argument to a function only by value and not by reference.

Q23. How an object is serialized in java?

Ans: In java, to convert an object into byte stream by serialization, an interface with the name Serializable is implemented by the class. All objects of a class implementing serializable interface get serialized and their state is saved in byte stream.

Q24. When we should use serialization?

Ans: Serialization is used when data needs to be transmitted over the network. Using serialization, object's state is saved and converted into byte stream .The byte stream is transferred over the network and the object is re-created at destination.

Q25. Is it compulsory for a Try Block to be followed by a Catch Block in Java for Exception handling?

Ans: Try block needs to be followed by either Catch block or Finally block or both. Any exception thrown from try block needs to be either caught in the catch block or else any specific tasks to be performed before code abortion are put in the Finally block.

Q26. Is there any way to skip Finally block of exception even if some exception occurs in the exception block?

Ans: If an exception is raised in Try block, control passes to catch block if it exists otherwise to finally block. Finally block is always executed when an exception occurs and the only way to avoid execution of any statements in Finally block is by aborting the code forcibly by writing following line of code at the end of try block:

```
System.exit(0);
```

Q27. When the constructor of a class is invoked?

Ans: The constructor of a class is invoked every time an object is created with new keyword.

For example, in the following class two objects are created using new keyword and hence, constructor is invoked two times.

```
public class const_example {  
  
    const_example() {  
  
        system.out.println("Inside constructor");  
  
    }  
  
    public static void main(String args[]) {  
  
        const_example c1=new const_example();  
  
        const_example c2=new const_example();  
  
    }  
}
```

Q28. Can a class have multiple constructors?

Ans: Yes, a class can have multiple constructors with different parameters. Which constructor gets used for object creation depends on the arguments passed while creating the objects.

Q29. Can we override static methods of a class?

Ans: We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation (not at runtime). Even if we try to override static method, we will not get an compilation error, nor the impact of overriding when running the code.

Q30. In the below example, what will be the output?

```
public class superclass {
```

```
public void displayResult() {  
  
    system.out.println("Printing from superclass");  
  
}  
  
}  
  
public class subclass extends superclass {  
  
    public void displayResult() {  
  
        system.out.println("Displaying from subClass");  
  
        super.displayResult();  
  
    }  
  
    public static void main(String args[]) {  
  
        subclass obj=new subclass();  
  
        obj.displayResult();  
  
    }  
  
}
```

Ans: Output will be:

Displaying from subclass

Displaying from superclass

Q31. Is String a data type in java?

Ans: String is not a primitive data type in java. When a string is created in java, it's actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object.

Q32. In the below example, how many String Objects are created?

```
String s1="I am Java Expert";
```

```
String s2="I am C Expert";
```

```
String s3="I am Java Expert";
```

Ans: In the above example, two objects of Java.Lang.String class are created. s1 and s3 are references to same object.

Q33. Why Strings in Java are called as Immutable?

Ans: In java, string objects are called immutable as once value has been assigned to a string, it can't be changed and if changed, a new object is created.

In below example, reference str refers to a string object having value "Value one".

```
String str="Value One";
```

When a new value is assigned to it, a new String object gets created and the reference is moved to the new object.

```
str="New Value";
```

Q34. What's the difference between an array and Vector?

Ans: An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types.

Q35. What is multi-threading?

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

Q36. Why Runnable Interface is used in Java?

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

Q37. What are the two ways of implementing multi-threading in Java?

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

Q38. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?

Ans: Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it's always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

Q39. What's the purpose of using Break in each case of Switch Statement?

Ans: Break is used after each case (except the last one) in a switch so that code breaks after the valid case and doesn't flow in the proceeding cases too.

If break isn't used after each case, all cases after the valid case also get executed resulting in wrong results.

Q40. How garbage collection is done in Java?

Ans: In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method.

Q41. How we can execute any code even before main method?

Ans: If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method.

Q42. Can a class be a super class and a sub-class at the same time? Give example.

Ans: If there is a hierarchy of inheritance used, a class can be a super class for another class and a sub-class for another one at the same time.

In the example below, continent class is sub-class of world class and it's super class of country class.

```
public class world {
```

```
.....
```

```
}
```

```
public class continent extends world {
```

```
.....
```

{

```
public class country extends continent {
```

```
.....
```

{}**Q43. How objects of a class are created if no constructor is defined in the class?**

Ans: Even if no explicit constructor is defined in a java class, objects get created successfully as a default constructor is implicitly used for object creation. This constructor has no parameters.

Q44. In multi-threading how can we ensure that a resource isn't used by multiple threads simultaneously?

Ans: In multi-threading, access to the resources which are shared among multiple threads can be controlled by using the concept of synchronization. Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it has become free from the other one using it.

Q45. Can we call the constructor of a class more than once for an object?

Ans: Constructor is called automatically when we create an object using new keyword. It's called only once for an object at the time of object creation and hence, we can't invoke the constructor again for an object after its creation.

Q46. There are two classes named classA and classB. Both classes are in the same package. Can a private member of classA can be accessed by an object of classB?

Ans: Private members of a class aren't accessible outside the scope of that class and any other class even in the same package can't access them.

Q47. Can we have two methods in a class with the same name?

Ans: We can define two methods in a class with the same name but with different number/type of parameters. Which method is to get invoked will depend upon the parameters passed.

For example in the class below we have two print methods with same name but different parameters. Depending upon the parameters, appropriate one will be called:

```
public class methodExample {
```

```
    public void print() {
```

```
system.out.println("Print method without parameters.");  
}  
  
public void print(String name) {  
  
system.out.println("Print method with parameter");  
}  
  
public static void main(String args[]) {  
  
methodExample obj1= new methodExample();  
  
obj1.print();  
  
obj1.print("xx");  
}  
}
```

Q48. How can we make copy of a java object?

Ans: We can use the concept of cloning to create copy of an object. Using clone, we create copies with the actual state of an object.

Clone() is a method of Cloneable interface and hence, Cloneable interface needs to be implemented for making object copies.

Q49. What's the benefit of using inheritance?

Ans: Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility) is another great benefit which allow new functionality to be introduced without effecting existing derived classes.

Q50. What's the default access specifier for variables and methods of a class?

Ans: Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package,not outside the package.

Q51. Give an example of use of Pointers in Java class.

Ans: There are no pointers in Java. So we can't use concept of pointers in Java.

Q52. How can we restrict inheritance for a class so that no class can be inherited from it?

Ans: If we want a class not to be extended further by any class, we can use the keyword **Final** with the class name.

In the following example, Stone class is Final and can't be extend

```
public Final Class Stone {  
    // Class methods and Variables  
}
```

Q53. What's the access scope of Protected Access specifier?

Ans: When a method or a variable is declared with Protected access specifier, it becomes accessible in the same class,any other class of the same package as well as a sub-class.

Access Levels				
Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Q54. What's difference between Stack and Queue?

Ans: Stack and Queue both are used as placeholder for a collection of data. The primary difference between a stack and a queue is that stack is based on Last in First out (LIFO) principle while a queue is based on FIFO (First In First Out) principle.

Q55. In java, how we can disallow serialization of variables?

Ans: If we want certain variables of a class not to be serialized, we can use the keyword **transient** while declaring them. For example, the variable trans_var below is a transient variable and can't be serialized:

```
public class transientExample {  
    private transient trans_var;  
    // rest of the code
```

{}

Q56. How can we use primitive data types as objects?

Ans: Primitive data types like int can be handled as objects by the use of their respective wrapper classes. For example, Integer is a wrapper class for primitive data type int. We can apply different methods to a wrapper class, just like any other object.

Q57. Which types of exceptions are caught at compile time?

Ans: Checked exceptions can be caught at the time of program compilation. Checked exceptions must be handled by using try catch block in the code in order to successfully compile the code.

Q58. Describe different states of a thread.

Ans: A thread in Java can be in either of the following states:

- Ready: When a thread is created, it's in Ready state.
- Running: A thread currently being executed is in running state.
- Waiting: A thread waiting for another thread to free certain resources is in waiting state.
- Dead: A thread which has gone dead after execution is in dead state.

Q59. Can we use a default constructor of a class even if an explicit constructor is defined?

Ans: Java provides a default no argument constructor if no explicit constructor is defined in a Java class. But if an explicit constructor has been defined, default constructor can't be invoked and developer can use only those constructors which are defined in the class.

Q60. Can we override a method by using same method name and arguments but different return types?

Ans: The basic condition of method overriding is that method name, arguments as well as return type must be exactly same as is that of the method being overridden. Hence using a different return type doesn't override a method.

Q61. What will be the output of following piece of code?

```
public class OperatorExample {  
  
    public static void main(String args[]) {  
  
        int x=4;  
  
        System.out.println(x++);  
  
    }  
}
```

{}

Ans: In this case postfix ++ operator is used which first returns the value and then increments. Hence it's output will be 4.

Q61. A person says that he compiled a java class successfully without even having a main method in it? Is it possible?

Ans: main method is an entry point of Java class and is required for execution of the program however; a class gets compiled successfully even if it doesn't have a main method. It can't be run though.

Q62. Can we call a non-static method from inside a static method?

Ans: Non-Static methods are owned by objects of a class and have object level scope and in order to call the non-Static methods from a static block (like from a static main method), an object of the class needs to be created first. Then using object reference, these methods can be invoked.

Q63. What are the two environment variables that must be set in order to run any Java programs?

Ans: Java programs can be executed in a machine only once following two environment variables have been properly set:

1. PATH variable
2. CLASSPATH variable

Q64. Can variables be used in Java without initialization?

Ans: In Java, if a variable is used in a code without prior initialization by a valid value, program doesn't compile and gives an error as no default value is assigned to variables in Java.

Q65. Can a class in Java be inherited from more than one class?

Ans: In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritances is not supported by Java.

Q66. Can a constructor have different name than a Class name in Java?

Ans: Constructor in Java must have same name as the class name and if the name is different, it doesn't act as a constructor and compiler thinks of it as a normal method.

Q67. What will be the output of Round(3.7) and Ceil(3.7)?

Ans: Round(3.7) returns 4 and Ceil(3.7) returns 4.

Q68: Can we use goto in Java to go to a particular line?

Ans: In Java, there is not goto keyword and java doesn't support this feature of going to a particular labeled line.

Q69. Can a dead thread be started again?

Ans: In java, a thread which is in dead state can't be started again. There is no way to restart a dead thread.

Q70. Is the following class declaration correct?

Ans:

```
public abstract final class testClass {  
    // Class methods and variables  
}
```

Ans: The above class declaration is incorrect as an abstract class can't be declared as Final.

Q71. Is JDK required on each machine to run a Java program?

Ans: JDK is development Kit of Java and is required for development only and to run a Java program on a machine, JDK isn't required. Only JRE is required.

Q72. What's the difference between comparison done by equals method and == operator?

Ans: In Java, equals() method is used to compare the contents of two string objects and returns true if the two have same value while == operator compares the references of two string objects.

In the following example, equals() returns true as the two string objects have same values. However == operator returns false as both string objects are referencing to different objects:

```
public class equalsTest {  
  
    public static void main(String args[]) {  
  
        String str1 = new String("Hello World");  
  
        String str2 = new String("Hello World");  
  
        if (str1.equals(str2))  
            { // this condition is true
```

```
System.out.println("str1 and str2 are equal in terms of values");

}

if (str1 == str2) {

    //This condition is true

    System.out.println("Both strings are referencing same object");

} else

{

    // This condition is NOT true

    System.out.println("Both strings are referencing different objects");

}

}
```

Q73. Is it possible to define a method in Java class but provide its implementation in the code of another language like C?

Ans: Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately.

Q74. How are destructors defined in Java?

Ans: In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced.

Q75. Can a variable be local and static at the same time?

Ans: No a variable can't be static as well as local at the same time. Defining a local variable as static gives compilation error.

Q76. Can we have static methods in an Interface?

Ans: Static methods can't be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java.

Q77. In a class implementing an interface, can we change the value of any variable defined in the interface?

Ans: No, we can't change the value of any variable of an interface in the implementing class as all variables defined in the interface are by default public, static and Final and final variables are like constants which can't be changed later.

Q78. Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?

Ans: Even though automatic garbage collection is provided by Java, it doesn't ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources.

So, garbage collection helps in reducing the chances of a program going out of memory but it doesn't ensure that.

Q79. Can we have any other return type than void for main method?

Ans: No, Java class main method can have only void return type for the program to get successfully executed.

Nonetheless , if you absolutely must return a value to at the completion of main method , you can use System.exit(int status)

Q80. I want to re-reach and use an object once it has been garbage collected. How it's possible?

Ans: Once an object has been destroyed by garbage collector, it no longer exists on the heap and it can't be accessed again. There is no way to reference it again.

Q81. In Java thread programming, which method is a must implementation for all threads?

Ans: Run() is a method of Runnable interface that must be implemented by all threads.

Q82. I want to control database connections in my program and want that only one thread should be able to make database connection at a time. How can I implement this logic?

Ans: This can be implemented by use of the concept of synchronization. Database related code can be placed in a method which has **synchronized** keyword so that only one thread can access it at a time.

Q83. How can an exception be thrown manually by a programmer?

Ans: In order to throw an exception in a block of code manually, **throw** keyword is used. Then this exception is caught and handled in the catch block.

```
public void topMethod(){  
try{  
excMethod();  
}catch(ManualException e){ }  
  
public void excMethod{  
String name=null;  
if(name == null){  
throw (new ManualException("Exception thrown manually "));  
}  
}
```

Q84. I want my class to be developed in such a way that no other class (even derived class) can create its objects. How can I do so?

Ans: If we declare the constructor of a class as private, it will not be accessible by any other class and hence, no other class will be able to instantiate it and formation of its object will be limited to itself only.

Q85. How objects are stored in Java?

Ans: In java, each object when created gets a memory space from a heap. When an object is destroyed by a garbage collector, the space allocated to it from the heap is re-allocated to the heap and becomes available for any new objects.

Q86. How can we find the actual size of an object on the heap?

Ans: In java, there is no way to find out the exact size of an object on the heap.

Q87. Which of the following classes will have more memory allocated?

Class A: Three methods, four variables, no object

Class B: Five methods, three variables, no object

Ans: Memory isn't allocated before creation of objects. Since for both classes, there are no objects created so no memory is allocated on heap for any class.

Q88. What happens if an exception is not handled in a program?

Ans: If an exception is not handled in a program using try catch blocks, program gets aborted and no statement executes after the statement which caused exception throwing.

Q89. I have multiple constructors defined in a class. Is it possible to call a constructor from another constructor's body?

Ans: If a class has multiple constructors, it's possible to call one constructor from the body of another one using **this()**.

Q90. What's meant by anonymous class?

Ans: An anonymous class is a class defined without any name in a single line of code using new keyword.

For example, in below code we have defined an anonymous class in one line of code:

```
public java.util.Enumeration testMethod()
{
    return new java.util.Enumeration()
{
    @Override
    public boolean hasMoreElements()
    {
        // TODO Auto-generated method stub
        return false;
    }
    @Override
    public Object nextElement()
    {
        // TODO Auto-generated method stub
        return null;
    }
}
```

Q91. Is there a way to increase the size of an array after its declaration?

Ans: Arrays are static and once we have specified its size, we can't change it. If we want to use such collections where we may require a change of size (no of items), we should prefer vector over array.

Q92. If an application has multiple classes in it, is it okay to have a main method in more than one class?

Ans: If there is main method in more than one classes in a java application, it won't cause any issue as entry point for any application will be a specific class and code will start from the main method of that particular class only.

Q93. I want to persist data of objects for later use. What's the best approach to do so?

Ans: The best way to persist data for future use is to use the concept of serialization.

Q94. What is a Local class in Java?

Ans: In Java, if we define a new class inside a particular block, it's called a local class. Such a class has local scope and isn't usable outside the block where its defined.

Q95. String and StringBuffer both represent String objects. Can we compare String and StringBuffer in Java?

Ans: Although String and StringBuffer both represent String objects, we can't compare them with each other and if we try to compare them, we get an error.

Q96. Which API is provided by Java for operations on set of objects?

Ans: Java provides a Collection API which provides many useful methods which can be applied on a set of objects. Some of the important classes provided by Collection API include ArrayList, HashMap, TreeSet and TreeMap.

Q97. Can we cast any other type to Boolean Type with type casting?

Ans: No, we can neither cast any other primitive type to Boolean data type nor can cast Boolean data type to any other primitive data type.

Q98. Can we use different return types for methods when overridden?

Ans: The basic requirement of method overriding in Java is that the overridden method should have same name, and parameters. But a method can be overridden with a different return type as long as the new return type extends the original.

For example , method is returning a reference type.

Class B extends A{

```
A method(int x){
```

```
//original method
```

```
}
```

```
B method(int x){
```

```
//overridden method
```

```
}
```

```
}
```

Q99. What's the base class of all exception classes?

Ans: In Java, **Java.lang.Throwable** is the super class of all exception classes and all exception classes are derived from this base class.

Q100. What's the order of call of constructors in inheritance?

Ans: In case of inheritance, when a new object of a derived class is created, first the constructor of the super class is invoked and then the constructor of the derived class is invoked.

Guru99 Provides FREE ONLINE TUTORIAL on Various courses like

[Java](#)[MIS](#)[MongoDB](#)[BigData](#)[Cassandra](#)[Web Services](#)[SQLite](#)[JSP](#)[Informatica](#)[Accounting](#)[SAP Training](#)[Python](#)[Excel](#)[ASP Net](#)[HBase](#)[Project Management](#)[Test Management](#)[Business Analyst](#)[Ethical Hacking](#)[PMP](#)[Live Project](#)[SoapUI](#)[Photoshop](#)[Manual Testing](#)[Mobile Testing](#)[Data Warehouse](#)[R Tutorial](#)[Tableau](#)[DevOps](#)[AWS](#)[Jenkins](#)[Agile Testing](#)[RPA](#)[JUnit](#)[Software Engineering](#)[Selenium](#)[CCNA](#)[AngularJS](#)[NodeJS](#)[PLSQL](#)

**Stay updated with new
courses at Guru99
Join our Newsletter**

1) Explain microservices architecture

Microservice Architecture is an architectural development style which builds an application as a collection of small autonomous services developed for a business domain.

2) Name three commonly used tools for Microservices

- Wiremock, 2.) Docker and 3.) Hysrix are important Microservices tool.

3) What is Monolithic Architecture?

Monolithic architecture is like a big container in which all the software components of an application are clubbed inside a single package.

4) What are the advantages of microservices?

Here, are some significant advantages of using Microservices:

- Technology diversity, e., Microservices can mix easily with other frameworks, libraries, and databases
- Fault isolation, e., a process failure should not bring the whole system down.
- Greater support for smaller and parallel team
- Independent deployment
- Deployment time reduce

5) What is Spring Cloud?

Spring cloud is an Integration software that integrates with external systems. It allows microservices framework to build applications which perform restricted amounts of data processing.



6) Discuss uses of reports and dashboards in the environment of Microservices

Reports and dashboards help in monitoring and upkeep of Microservices. Tons of Application Monitoring Tools assist in this.

7) What are main differences between Microservices and Monolithic Architecture?

Microservices	Monolithic Architecture
Service Startup is fast	Service startup takes time
Microservices are loosely coupled architecture.	Monolithic architecture is mostly tightly coupled.
Changes done in a single data model does not affect other Microservices.	Any changes in the data model affect the entire database
Microservices focuses on products, not projects	Monolithic put emphasize over the whole project

8) What are the challenges faced while using Microservices?

- Microservices always rely on each other. Therefore, they need to communicate with each other.
- As it is distributed system, it is a heavily involved model.
- If you are using Microservice architecture, you need to ready for operations overhead.
- You need skilled professionals to support heterogeneously distributed microservices.

9) In which cases microservice architecture best suited?

Microservice architecture is best suited for desktop, web, mobile devices, Smart TVs, Wearable, etc.

10) Tell me the name of some famous companies which are using Microservice architecture

Most large-scale websites like Twitter, Netflix, Amazon, have advanced from a monolithic architecture to a microservices architecture.

11) What are the characteristics of Microservices?

- Essential messaging frameworks
- Decentralized Governance
- Easy Infrastructure automation
- Design for failure
- Infrastructure automation

12) What is RESTful?

Representational State Transfer (REST)/RESTful web services is an architectural style that helps computer systems to communicate over the internet. These web services make microservices easier to understand and implement.

13) Explain three types of Tests for Microservices? In Microservice architecture tests are divided into three broad categories:

- At the bottom level test, we can perform a general test like performance and unit tests. These kinds of tests are entirely automated.
- At the middle level, we can perform exploratory tests like the stress tests and usability tests.
- At the top level, we can conduct acceptance tests which are mostly fewer in numbers. It also helps stakeholders to know about different software features.

14) What are Client certificates?

Client certificates is a digital certificate used to make authenticated requests to a remote server. It is termed as a client certificate.

15) Explain the use of PACT in Microservices architecture?

It is an open source tool which allows testing interactions between service providers and consumers. However, it is separated from the contract made. This increases the reliability of the Microservices applications.

16) What is the meaning of OAuth?

OAuth means open authorization protocol. This protocol allows you to access the client applications on HTTP for third-party providers GitHub, Facebook, etc. It helps you to share resources stored on one site with another site without the need for their credentials.

17) What is End to End Microservices Testing?

End-to-end testing validates every process in the workflow is functioning correctly. It also ensures that the system works together as a whole and satisfies all requirements.

18) Why are Container used in Microservices?

Containers are easiest and effective method to manage the microservice based application. It also helps you to develop and deploy individually. Docker also allows you to encapsulate your microservice in a container image along with its dependencies. Microservice can use these elements without additional efforts.

19) What is the meaning of Semantic monitoring in Microservices architecture?

Semantic monitoring combines automated tests with monitoring of the application. It allows you to find out reasons why your business is not getting more profits.

20) What is a CDC?

CDC is Consumer-Driven Contract. It is a pattern for developing Microservices so that external systems can use them.

21) What is the use of Docker?

Docker offers a container environment which can be used to host any application. This software application and the dependencies that support it which are tightly-packaged together.

22) What are Reactive Extensions in Microservices?

Reactive Extensions is also called Rx. It is a design pattern which allows collecting results by calling multiple services and then compile a combined response. Rx is a popular tool in distributed systems which works exactly opposite to legacy flows.

23) Explain the term 'Continuous Monitoring.'

Continuous monitoring is a method which is used for searching compliance and risk issues associated with a company's operational and financial environment. It contains human, processes, and working systems which support efficient and actual operations.

24) How independent micro-services communicate with each other?

It depends upon your project needs. However, in most cases, developers use HTTP/REST with JSON or Binary protocol. However, they can use any communication protocol.

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

[Java](#)[MIS](#)[MongoDB](#)[BigData](#)[Cassandra](#)[Web Services](#)[SQLite](#)[JSP](#)[Informatica](#)[Accounting](#)[SAP Training](#)[Python](#)[Excel](#)[ASP Net](#)[HBase](#)

Project
Management

Test Management Business Analyst Ethical Hacking

PMP

Live Project

SoapUI

Photoshop

Manual Testing

Mobile Testing

Data Warehouse

R Tutorial

Tableau

DevOps

AWS

Jenkins

Agile Testing

RPA

JUnit

Software
Engineering

Selenium

CCNA

AngularJS

NodeJS

PLSQL

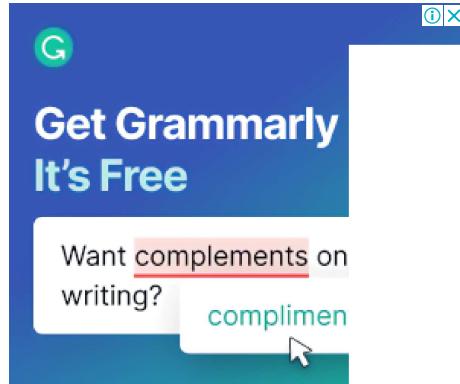
**Stay updated with new
courses at Guru99
Join our Newsletter**

Javarevisited

Blog about Java, Programming, Spring, Hibernate, Interview Questions, Books and Online Course Recommendations from Udemy, Pluralsight, Coursera, etc

Top 22 Spring Interview Questions Answers for Java Programmers [UPDATED]

Hey Java Programmers, if you are preparing for a Java developer interview then you should pay special attention to Spring framework-related questions. **Spring framework interview questions** are on the rise on Java web development and core Java interviews. This is obvious given Spring Framework is the best most popular framework available for Java application development and now *Spring IOC container* and Spring MVC framework are used as a de-facto framework for all new Java development. Because of its popularity, interview questions from the spring framework are top on any list of **Java Interview questions** and you should also prepare different Spring topics like Spring Boot, Spring Data JPA, Spring Security, Testing and Spring Cloud, etc to do well on Java developer interviews.



After some of my readers requested, I thought to put together some spring interview questions and answers which have appeared on many Java interviews and useful for practicing before appearing on any Java Job interview. I first wrote this article a long back and even its content is still relevant I thought to update it, especially after finishing my list of **Spring Boot Interview Questions** recently.

This list of Spring interview questions and answers contains questions from Spring fundamentals like Spring IOC and Dependency Injection, **Spring MVC Framework**, Spring Security, Spring AOP, etc, because of the length of this post I haven't included Spring interview questions from Spring JDBC and JMS which is also a popular topic in [core Java](#) and [Java EE interviews](#). I suggest preparing those as well.

Anyway, these Spring questions are not very difficult and based on fundamentals like what is the default scope of Spring bean? which is mostly asked during the first round or the telephonic round of Java interview. Although you can find answers to these **Spring interview questions** by doing Google I have also included answers for most of the questions for your quick reference.

As I have said before both Spring and Spring MVC, and now Spring Boot are fantastic Java frameworks and if you are not using them then it's a good time to start using them, these questions will give you some head start as well. Spring MVC can be used to develop both standalone Java web applications as well as RESTful Web Services using Spring.

If you are completely new to the Spring framework, I suggest you first take a look at **Spring Framework 5: Beginner to Guru** course, otherwise, most of these questions will not make any sense to you. It's also one of the most up-to-date courses which not only cover the basics of IOC and DI container but also advanced concepts like reactive programming with Spring 5.

Discover your IQ

test-iq.org

Top 22 Spring Framework Interview Questions and Answers

These articles cover Interview questions from different Spring Framework projects like Spring MVC, Core Spring, and [Spring with REST](#) and [Spring MVC](#) particularly. I have also listed resources and linked some articles to learn more and get some more practice questions if you are interested in further learning.

Now let's start with questions, these **Spring Interview Questions** are not very tricky or tough and based upon primary concepts of the spring framework.

If you are developing an application using the Spring framework then you may be, already familiar with many of these Java and Spring interview questions and answers. Nevertheless, it's a good recap before appearing in any Spring and Java interview.

1. Spring Core Interview Questions

Question 1: What is IOC or inversion of control? ([answer](#))

Answer: This *Spring interview question* is the first step towards the Spring framework and many interviewers start Spring interviews from this question. As the name implies **Inversion of the control** means now we have inverted the control of creating the object from our own using a new operator to container or framework.

Now it's the responsibility of the container to create an object as required. We maintain one XML file where we configure our components, services, all the classes, and their property. We just need to mention which service is needed by which component and container will create the object for us.

This concept is known as [dependency injection](#) because all object dependency (resources) is injected into it by the framework.

Example:

```
<bean id="createNewStock"
      class="springexample.stockMarket.CreateNewStockAccont">
    <property name="newBid"/>
</bean>
```

In this example, the `CreateNewStockAccont` class contains the getter and setter for `newBid`, and the container will instantiate `newBid` and set the value automatically when it is

Question 2: Explain the Spring Bean-LifeCycle.

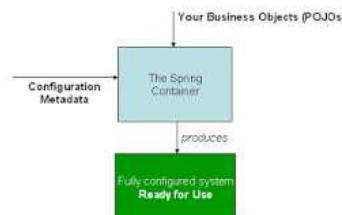
Ans: Spring framework is based on IOC so we call it an IOC container also So Spring beans reside inside the IOC container. Spring beans are nothing but Plain old java objects (POJO). The following steps explain their life cycle inside the container.

1. The container will look at the bean definition inside the configuration file (e.g. bean.xml).
- 2 using reflection container will create the object and if any property is defined inside the bean definition then it will also be set.
3. If the bean implements the `BeanNameAware` interface, the factory calls `setBeanName()` passing the bean's ID.
4. If the bean implements the `BeanFactoryAware` interface, the factory calls `setBeanFactory()`, passing an instance of itself.
5. If there are any `BeanPostProcessors` associated with the bean, their `postProcessBeforeInitialization()` methods will be called before the properties for the Bean are set.
6. If an `init()` method is specified for the bean, it will be called.
7. If the Bean class implements the `DisposableBean` interface, then the `destroy()` method will be called when the Application no longer needs the bean reference.
8. If the Bean definition in the Configuration file contains a 'destroy-method' attribute, then the corresponding method definition in the Bean class will be called.

These were just some of the Spring Fundamentals I can cover here if you are interested to learn more I suggest you take a look at [Spring Master Class - Beginner to Expert](#), an end-to-end course to learn Spring.

Spring IOC Containers

- How does an IoC container works?



Question 3: What is Bean Factory, have you used XMLBeanFactory?

Ans: BeanFactory is a factory Pattern that is based on IOC [design principles](#).it is used to make a clear separation between application configuration and dependency from actual code. The `XmlBeanFactory` is one of the implementations of Bean Factory which we have used in our project.

Bean instances defined in our XML file.

```
BeanFactory factory = new XmlBeanFactory(new FileInputStream("beans.xml"));
```

Or

```
ClassPathResource resorce = new ClassPathResource("beans.xml");
XmlBeanFactory factory = new XmlBeanFactory(resorce);
```

Question 4: What are the difference between BeanFactory and ApplicationContext in Spring? [\(answer\)](#)

Answer: This one is a very popular Spring interview question and often asks in an entry-level interview. `ApplicationContext` is the preferred way of using spring because of the functionality provided by it and the interviewer wanted to check whether you are familiar with it or not.

ApplicationContext.	BeanFactory
Here we can have more than one config files possible	In this only one config file or .xml file
Application contexts can publish events to beans that are registered as listeners	Don't support.
Support internationalization (I18N) messages	It's not
Support application life-cycle events, and validation.	Doesn't support.
Supports many enterprise services such as JNDI access, EJB integration, remoting	Doesn't support.

Question 5: What is AOP?

Answer: The core construct of AOP is the aspect, which encapsulates behaviors affecting multiple classes into reusable modules. AOP is a programming technique that allows a developer to modularize crosscutting concerns, that cut across the typical divisions of responsibility, such as **logging and transaction management**.

Spring AOP, aspects are implemented using regular classes or regular classes annotated with the `@Aspect` annotation. You can also check out these [30+ Spring MVC interview questions](#) for more focus on Java web development using the Spring MVC framework.

Question 6: Explain Advice?

Answer: It's an implementation of aspect; advice is inserted into an application at join points. Different types of advice include "around," "before" and "after" advice

Question 7: What are the joint Point and point cut?

A join point is an opportunity within the code for which we can apply an aspect. In Spring AOP, a join point always represents a method execution.

Pointcut: a predicate that matches join points. A pointcut is something that defines what join-points advice should be applied.

If you need more Spring AOP questions then you can also check out my article about [17 Spring AOP Interview Questions with Answers](#) for Java developers.

Question 8: Difference between the setter and constructor injection in Spring? ([answer](#))

Setter injection is more flexible than constructor injection because you must remember the type and order of the constructor parameters. Also, constructor injection is generally used to inject the mandatory dependency, while setter can be used to inject the optional dependency.

Question 9: Difference between Factory Pattern and Dependency Injection in Java? ([answer](#))

Even though both allow you to reduce coupling in code, dependency injection is much more flexible and easier to test than Factory pattern.

Questions 10. Difference between @Autowired and @Inject annotation in Spring?

([answer](#))

Question 11: What are the different modules in spring?

Answer: spring has seven core modules

1. The Core container module
2. Application context module
3. AOP module (Aspect Oriented Programming)
4. JDBC abstraction and DAO module
5. O/R mapping integration module (Object/Relational)
6. Web module
7. MVC framework module

2. Spring MVC Interview Questions Answers

So far we have seen the Spring core interview questions and answers and now let's see some interview questions from Spring MVC, one of the most important parts of Spring framework which allows you to develop web applications in Java using Model View Controller design pattern.

Questions 12: What is the difference between @Controller and @RestController in Spring MVC? ([answer](#))

Even though both are used to indicate that a Spring bean is a Controller in Spring MVC setup, `@RestController` is better when you are developing [RESTful web services](#) using the Spring MVC framework. It's a combination of `@Controller` + `@ResponseBody` annotation which allows the controller to directly write the response and bypassing the view resolution process, which is not required for RESTful web service.

It also instructs `DispatcherServlet` to use different `HttpMessageConverters` to represent the response in the format client is expecting e.g. `HttpMessageJackson2Convert` to represent response in JSON format and JAXB based message converts to generate XML response.

You can further see the [REST with Spring](#) course by Baeldung to learn more about developing

```

13 import springfox.documentation.builders.RequestHandlerSelectors;
14 import springfox.documentation.spi.DocumentationType;
15 import springfox.documentation.spring.web.plugins.Docket;
16 import springfox.documentation.swagger2.annotations.EnableSwagger2;
17
18 @Configuration
19 @ComponentScan({ "org.baeldung.common.web", "org.baeldung.um.web" })
20 @EnableWebMvc
21 @EnableSwagger2
22 public class UmWebConfig extends WebMvcConfigurerAdapter {
23
24     public UmWebConfig() {
25         super();
26     }
27
28     // Beans
29
30     @Bean
31     public Docket mainConf() { /* @formatter:off */
32         return new Docket(DocumentationType.SWAGGER_2)
33             .select().apis(RequestHandlerSelectors.any())
34             .paths(PathSelectors.any())
35             .apiInfo(apiInfo)
36             .apiMapping("/*")
37             .directModelSubstitute(LocalDate.class, String.class)
38             .genericModelSubstitutes(ResponseEntity.class)
39             ./* @formatter:on */
40     }
41 }
```

Question 13: What is the difference between a singleton and a prototype bean?

Ans: This is another popular *spring interview question* and an important concept to understand. Basically, a bean has scopes which define their existence on the application.

Singleton: means single bean definition to a single object instance per Spring IOC container.

Prototype: means a single bean definition to any number of object instances.

Whatever beans we defined in the spring framework are singleton beans.

There is an attribute in the bean tag named ‘singleton’ if specified true then the bean becomes singleton and if set to false then the bean becomes a prototype bean. By default, it is set to `true`. So, all the beans in the spring framework are by default singleton beans.

```

<bean id="createNewStock" class="springexample.stockMarket.CreateNewStockAccount"
      singleton="false">
    <property name="newBid"/>
</bean>
```

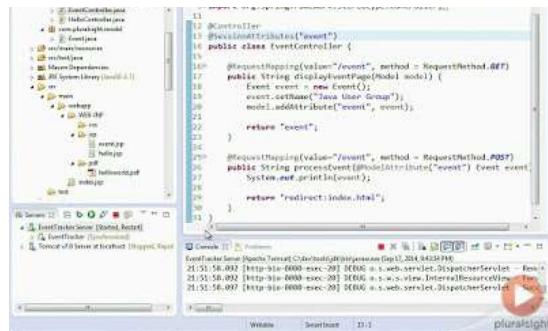
Question 14: What is the role of DispatcherServlet in Spring MVC? ([answer](#))

The `DispatcherServlet` is very important from the Spring MVC perspective, it acts as a `FrontController` i.e. all requests pass through it. It is responsible for routing the request to the controller and view the resolution before sending the response to the client.

When Controller returns a Model or View object, it consults all the view resolvers registered to find the correct type of `ViewResolver` which can render the response for clients.

In the case of RESTful Web Services, the `DispatcherServlet` is also responsible for using `HttpMessageConverter` to represent the response in the JSON, XML, or TEXT format, depending on the content negotiation between Client and Server like if the client sends a request with HTTP accept header as “`application/json`” then `DispatcherServlet` will ask the `HttpMessageJackson2Converter` to convert the response into JSON format.

You can further see the free [Introduction to Spring MVC](#) course from Pluralsight to learn more about Spring MVC and `DispatcherServlet`.



Question 15: How to call the stored procedure from Java using Spring Framework?

[\(answer\)](#)

Question 16: How to Setup JDBC Database connection pool in Spring Web application?

[\(answer\)](#)

Questions 17: Difference between @RequestParam and @PathVariable in Spring MVC?

[\(answer\)](#)

Questions 18: Difference between @Component, @Service, @Controller, and @Repositoring annotation in Spring MVC? [\(answer\)](#)

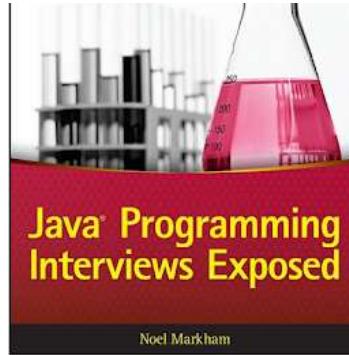
Question 19: What type of transaction Management Spring support?

Ans: This spring interview question is a little difficult as compared to previous questions just because **transaction management** is a complex concept and not every developer familiar with it. Transaction management is critical in any application that will interact with the database.

The application has to ensure that the data is consistent and the integrity of the data is maintained. Following two types of transaction management is supported by spring:

1. Programmatic transaction management
2. Declarative transaction management.

If you need more questions, you can also check out the [Java Programming Interview exposed](#) book from Wrox publication, apart from various Java topics it also contains some really good Spring framework, Hibernate, and Spring MVC questions with detailed explanation.



Also, Hibernate is mostly used in Spring, so don't forget to prepare some [Hibernate interview questions](#) along with Spring.

3. Spring Security Interview Questions Answer

Some of the readers requested to provide Spring Security interview questions and answer, So I thought to update this article with a few of the Spring security questions I came across.

Here are those:

20. How do you control the concurrent Active session using Spring Security? ([answer](#))

Another Spring interview question is based on Out of box feature provided by the Spring framework. You can easily control How many active sessions a user can have with a Java application by using Spring Security.

Apart from that Spring Security also provides the "remember me" feature which you can use to provide easier access for your users by remembering their login details on their personal computer. You can further see [Learn Spring Security](#) course by Eugen Paraschiv to learn more about advanced details of Spring Security.

21. How do you set up LDAP Authentication using Spring Security? ([answer](#))

This is a very popular Spring Security interview question as Spring provides out-of-the-box support to connect Windows Active Directory for LDAP authentication and with few configurations in the Spring config file you can have this feature enabled.

22. How to implement Role-Based Access Control (RBAC) using Spring Security? ([answer](#))

Spring Security provides a couple of ways to implement Role-based access control like by using GrantedAuthority. See the article to learn more about it.

That's all about frequently asked Spring Interview Questions with Answers. These **Spring interview Questions and answers** are not very difficult and focused on spring fundamentals rather than focusing on an advanced feature of session management, spring security, authentication, etc. we will cover those questions in some other interview article. I would also suggest that share some spring questions asked to you guys during the interview and then I can put together those with their answers for quick reference of everybody.

Other **Spring Articles and Interview Questions** you may like

- [Spring Boot Features Java programmer Should know](#)
- [10 Frameworks Java and Web Developer Should learn](#)
- [Top 5 Spring Microservice Courses for Java developers](#)
- [3 Best Practices Java Devs can Learn from Spring](#)
- [10 Tips to become a better Java Programmer](#)
- [5 Courses to Learn Spring Boot for Beginners](#)
- [5 Spring Books Experienced Java Developers Should Read](#)
- [10 Spring MVC Annotations Java developer should know](#)
- [20+ Spring Boot Interview Questions with Answers](#)
- [15 Microservice Interview Question with Answers](#)
- [20 Spring Boot Testing Interview Questions](#)
- [15+ Spring Data JPA Interview Questions with Answers](#)
- [15+ Spring Cloud Interview Questions with Answers](#)
- [25+ Spring Security Interview Questions in Java](#)

Thanks for reading this article, if you like this article then please share it with your friends and colleagues.

P. S. - If you are preparing for Java Interviews and need more Spring Framework Interview questions for Practice then you can also check out this [Spring Framework Interview Guide - 200+ Questions & Answers](#), which contains 200+ real-world questions and their explanations.



DOWNLOAD YOUR FREE JAVA

Join my list for exclusive discounts, coupons, and early access to my books and courses.

And, download a free chapter from my best selling Grokking the Java interview book when you sign up

Email Address

Download

We respect your privacy. Unsubscribe at any time.

By javin paul 

Labels: [core java interview question](#), [jsp-servlet](#), [spring](#), [spring interview questions](#)

47 comments :

Anonymous said...

Spring is largely a redundancy. Java EE 6 provides what you are likely to need from Spring a standards compliant framework.

If you are still doing J2EE development I can understand why you think Spring is the better framework, however.

October 9, 2011 at 7:27 AM

[Javin @ java OutOfMemoryError](#) said...

@Anonymous, Thanks for your comments, In My opinion Spring is better than J2EE , though this is a premature comment because I have not used J2EE quite except Jsp , Servlet and EJB a lot but with that limited experience I found Spring more better.

October 10, 2011 at 5:30 AM

[Javin @ Java Generic tutorial](#) said...

Thanks Thomas you like these spring questions, and thanks for explaining that point some how I completely missed that.

October 13, 2011 at 5:54 AM

Raja said...

Can you add some more Spring framework interview questions , especially from spring mvc part. I am going for spring interview and expecting spring mvc interview questions based on job description. Also Can you please upload pdf version of spring interview questions so that I can take a print out.

November 29, 2011 at 8:40 PM

Raja said...

Also it would have been little better if you could split up these spring interview question on different category e.g. Spring IOC interview questions, Spring AOP interview questions, Spring MVC interview questions, Spring security interview questions , Spring dependency injection interview questions etc.

November 29, 2011 at 8:55 PM

Sarabdeep said...

I think Spring is awesome framework and very modular. One of best frameworks around in Open Source Projects. I am just overwhelmed with features and abilities provided by these framework. I don't think it needs anyone approval. Most of bigger firms use it in bigger projects for scalability. Love Live Spring Framework.....

December 11, 2011 at 7:51 PM

Anonymous said...

these spring interview questions are rather easy, you should include touch spring questions doesn't matter whether it appeared on interview or not.

January 5, 2012 at 7:46 PM

Gaurav said...

can you also put some spring mvc questions, like questions on controller, dispatch-servlet.xml and view mappings on spring framework. spring-mvc is more asked than any other module of spring framework so please share some spring

Anonymous said...

@Raja, did you get the job you were interviewing for? Doubt it since you can't cut and paste the Q & A in this article and print them on your own. Please make me a pdf...whaaaa

February 14, 2012 at 6:24 PM

Priyanshu Joshi said...

Here are few more challenging and tough Spring interview question:

- 1) What is Spring IOC Container ?
- 2) Does Singleton from Spring Container is thread safe?
- 3) Why Spring MVC is better than Struts?
- 4) What is Spring Integration?
- 5) How to call remote method by RMI using Spring
- 6) What scheduling feature Spring framework provides?
- 7) How do you make a Singleton bean to lazy load in ApplicationContext which loads all Singleton beans eagerly during startup?
- 8) Does Spring Security part of Spring framework?
- 9) How to configure Spring using Annotation
- 10) Which version of Spring have you used recently and what is difference you observed from previous spring version.

April 24, 2012 at 6:52 PM

Saumya said...

@Priyanshu

Answer of "What is Spring IOC Container"

Spring IOC container is an implementation of Dependency Injection design pattern provided by Spring framework.

Instead of creating objects in application code using new() keyword, we use

ApplicationContext.getBean("nameOfBean") to get reference of Objects. Spring framework creates objects, manages object and set the dependency automatically so that you can get Object in proper state.

Answer 2 - "Does Spring Singleton are thread-safe"

I am not sure on this , I guess this may depends upon whether object is immutable or not.

May 21, 2012 at 7:25 PM

Sneha said...

Hi, I am an experienced Java programmer (4 years experience in Java and J2EE technology) but I am new to Spring framework. I am going to appear for client interview this week and looking for some Spring questions for 2 to 4 years experienced programmer. Would preparing above Spring question would be sufficient? if possible can you please share some tough, difficult or triky Spring interview question as most of them are quite command and frequently asked. They are also looking for Some spring MVC , Spring JDBC and Spring Security experience. Would be great if you could share few Spring MVC, JDBC and Spring Security interview question and answers as well.

Thanks you

September 4, 2012 at 9:29 PM

Jeetu said...

Can you please let me know How can we download this Spring interview question answers as pdf ?

September 4, 2012 at 9:30 PM

Vikash said...

Please share some questions answers for Spring hibernate integration and Spring AOP support. Also if you could include questions like What is new in Spring 3.0, Which new features are introduced in Spring 3.0 etc, That would be fantastic.

September 4, 2012 at 9:32 PM

Anonymous said...

Few of Spring JDBC interview questions, which I am aware of:

What is benefit of using JdbcTemplate ? Why should you use JdbcTemplate in Spring ?

How do you handle SQLException in Spring?

How to you setup JDBC connection pool in Spring etc.

interview questions from JDBC in Spring is not that many, but you should be familiar with JdbcTemplate.

September 6, 2012 at 10:01 PM

Anonymous said...

Spring is light weight component and open source.Dependency injection based so no need to configure the class file in *.xml

December 26, 2012 at 3:45 AM

sunil choudhury said...

can it be possible to get the bean instance without using getBean() not even for a single time.

reards,
sunil choudhury

March 17, 2013 at 1:14 AM

Javin @ Singleton vs Static class Java said...

Hi Sunil, IOC does lot of stuff for you. It creates dependency and construct objects. If you don't need explicit instance of bean, you can avoid that getBean() call, but you almost always need that. It's better to wrap casting and call to getBean in a separate class itself and use Generics , instead of casting everywhere in code.

March 18, 2013 at 6:00 AM

Anonymous said...

question about prototype was asked to me in ML interview. The tougher question is why we need a prototype scope ?
the answer is if you want one instance of the bean per thread or session i.e. if you need a stateful bean

May 4, 2013 at 5:00 PM

Anonymous said...

What is the difference between Spring MVC and struts2 and which one is better to use

June 28, 2013 at 4:14 AM

Unknown said...

Sir

I have .war file in java eclipse, but how can i upload it on server please tell me
because when i am asking to the domain giver sites they recommend me
to convert it into either asp.net or any other html files.

Will you please give me result???

November 5, 2013 at 11:54 PM

Unknown said...

Hi,

Dear sir

I have a query, I have coded a web site in jee, html, js
in eclipse and I got a .war file in it and I want to upload
it on server as the same, I have asked to many domain giver
how to upload it in .war file but they recommended me to use it either
in asp.net or in completely html form
Please Tell me how can I do it???

November 6, 2013 at 12:00 AM

Unknown said...

Arpan....you are so confusing and your english is funny. Some needs pdf's some needs helpguys do your homework
and work hard ..there is no free lunch. Don't be so stupid!!...you are a shame to IT

April 6, 2014 at 6:01 AM

Anonymous said...

@sskcar...Here Peoples ask for help, if you can't provide any help to them then please , you don't have right to say
whatever u said in your recent comments.

April 8, 2014 at 11:18 AM

Guruprasad said...

Hi,

I recently attended a Spring interview and got a follow up question on Question 6.

How would you inject a prototype bean into a singleton bean from a method?

I answered that I would create the prototype bean using the application context giving the bean id inside the method
of the singleton class.

But apparently there is another way of doing this:

1. Use element when creating the Singleton class
2. using the element when creating the Prototype class.

May 1, 2014 at 1:38 AM

Anonymous said...

August 5, 2014 at 1:43 AM

Unknown said...

I am nikita. Currently working in bpo company. Having 1 year experience.
I want my career in java.have completed 1 project in java.
Can I switch into java bcz I am from computer background.
In java wants 3-4 year experience. And now I can't apply fresher also.
What should I do?
Anyone plz suggest me..

September 14, 2014 at 8:18 AM

Unknown said...

I have one question
I am currently working in bpo company.
But I want career in java.
If I complete Project in spring .then is it possible to get hire in IT company? Without having any experience in java?

I am from computer background

September 14, 2014 at 8:27 AM

javin paul said...

Hello Nikita, If you have any engineering degree or courses on computer science e.g. M.Sc or MCA, it's quite easy to get Job on Java Software development. I also suggest to do Oracle Java Certification, a good score there will boost your chances of Java job for sure. Coming to Spring, it's not mandatory for Java, but having this experience is always good. Good luck.

September 14, 2014 at 9:30 PM

Unknown said...

I have completed bcs .no any certification in java .I thought with spring I can enter into java and Will no wastage of experience.

September 14, 2014 at 9:55 PM

javin paul said...

I presume, BCS is a bachelors degree in computer science, so that's good. If you don't have any experience than I suggest to do certification and at same time try to work as intern or trainee to gain some experience. Having said that, no harm in trying and giving Job interviews. There are times when company needs developer in large numbers and they are less restrictive in filtering. Having worked in Spring will also help you.

September 14, 2014 at 11:13 PM

Unknown said...

Thanks Javin for better suggestions.I
Will prefer this one. One more question which certification would be best ocjp or scjp?

September 15, 2014 at 11:03 AM

javin paul said...

Your Welcome Nikita, Go for OCJP, it's new name of SCJP after Oracle acquired Sun in 2010

September 16, 2014 at 6:22 AM

Unknown said...

Hi Guruprasad,

I still dont understand your point. Could you pls give us an example?

November 2, 2014 at 11:47 PM

Anonymous said...

All beans created in Spring are by default Singleton but not thread safe.

December 18, 2014 at 6:00 AM

Anonymous said...

hello
i want my career in java.i have no experience .so pls suggest me .how get job in java developer.

April 17, 2015 at 7:26 PM

Anonymous said...

Hi,
Thanks for such an useful topic.

April 20, 2015 at 7:34 PM

Anonymous said...

Very useful questions on Spring, for more Java interview questions from last 5 years, You can also see this list of [latest Java questions](#)

October 21, 2015 at 7:36 PM

Unknown said...

How to explain spring mvc along with your current project online and what are the module should i explain plz help me out

November 25, 2015 at 8:02 AM

Unknown said...

how to deal with cyclic dependency in spring config xml is also a popular question asked to me & one of my friend too..

Bean A --->BeanB

BeanB ---> BeanA

December 1, 2015 at 8:34 AM

Anonymous said...

Can you include some questions/answers on spring batch topic

April 22, 2016 at 12:07 AM

Unknown said...

Without learning j2EE we can understand spring and hibernate concept.

June 25, 2016 at 7:29 AM

javin paul said...

@chandan Ons, you are correct now there is no need to learn Java EE to use Spring or Hibernate, you can even use them in core Java project which requires database access. Spring is now mainly use for DI and Hibernate is for ORM.

June 25, 2016 at 6:55 PM

Unknown said...

Can you please explain some basic benefits of Dependency Injection ?

February 1, 2017 at 5:07 AM

Jagadeesh said...

for your 3rd question answer..

do we have singleton="false/true" attribute in bean definition ? I think it is scope attribute.

April 20, 2017 at 11:15 AM

harsh said...

what is dispatcher servlet

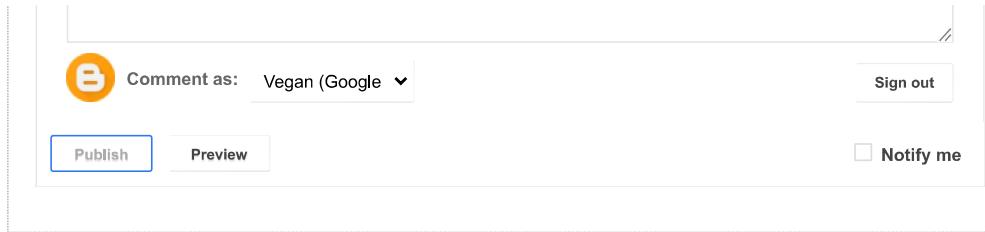
September 12, 2019 at 12:22 PM

javin paul said...

It's Servlet implementation in Spring MVC which works as front controller. All the request to your web application goes through this class in spring mvc application.

September 14, 2019 at 2:10 AM

Post a Comment

[Newer Post](#)[Home](#)[Older Post](#)Subscribe to: [Post Comments \(Atom \)](#)

Search This Blog

Spring Boot Interview Questions



Spring Boot is a Spring module which provides RAD (Rapid Application Development) feature to Spring framework.

It is used to create stand alone spring based application that you can just run because it needs very little spring configuration.

For more information [click here](#).

2) What are the advantages of Spring Boot?

- Create stand-alone Spring applications that can be started using java -jar.
- Embed Tomcat, Jetty or Undertow directly. You don't need to deploy WAR files.
- It provides opinionated 'starter' POMs to simplify your Maven configuration.
- It automatically configures Spring whenever possible.

For more information [click here](#).

3) What are the features of Spring Boot?

- Web Development
- SpringApplication
- Application events and listeners
- Admin features

For more information [click here](#).

4) How to create Spring Boot application using Maven?

There are multiple approaches to create Spring Boot project. We can use any of the following approach to create application.

- Spring Maven Project
- Spring Starter Project Wizard
- Spring Initializr

- o Spring Boot CLI



It is a web tool which is provided by Spring on official site. You can create Spring Boot project by providing project details.

For more information [click here](#).

6) How to create Spring Boot project using boot CLI?

It is a tool which you can download from the official site of Spring Framework. Here, we are explaining steps.

Download the CLI tool from official site and For more information [click here](#).

7) How to create simple Spring Boot application?

To create an application. We are using STS (Spring Tool Suite) IDE and it includes the various steps that are explaining in steps.

For more information [click here](#).

8) What are the Spring Boot Annotations?

The @RestController is a stereotype annotation. It adds @Controller and @ResponseBody annotations to the class. We need to import org.springframework.web.bind.annotation package in our file, in order to implement it.

For more information [click here](#).

9) What is Spring Boot dependency management?

Spring Boot manages dependencies and configuration automatically. You don't need to specify version for any of that dependencies.

Spring Boot upgrades all dependencies automatically when you upgrade Spring Boot.

For more information [click here](#).

10) What are the Spring Boot properties?

Spring Boot provides various properties which can be specified inside our project's



For more information [click here](#).

11) What are the Spring Boot Starters?

Starters are a set of convenient dependency descriptors which we can include in our application.

Spring Boot provides built-in starters which makes development easier and rapid. For example, if we want to get started using Spring and JPA for database access, just include the **spring-boot-starter-data-jpa** dependency in your project.

For more information [click here](#).

12) What is Spring Boot Actuator?

Spring Boot provides actuator to monitor and manage our application. Actuator is a tool which has HTTP endpoints. When application is pushed to production, you can choose to manage and monitor your application using HTTP endpoints.

For more information [click here](#).

13) What is thymeleaf?

It is a server side Java template engine for web application. Its main goal is to bring elegant natural templates to your web application.

It can be integrated with Spring Framework and ideal for HTML5 Java web applications.

For more information [click here](#).

14) How to use thymeleaf?

In order to use Thymeleaf we must add it into our pom.xml file like:

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

For more information [click here](#).



15) How to connect Spring Boot to the database using JPA?

For more information [click here](#).

16) How to connect Spring Boot application to database using JDBC?

Spring Boot provides starter and libraries for connecting to our application with JDBC. Here, we are creating an application which connects with Mysql database. It includes the following steps to create and setup JDBC with Spring Boot.

For more information [click here](#).

17) What is @RestController annotation in Spring Boot?

The **@RestController** is a stereotype annotation. It adds **@Controller** and **@ResponseBody** annotations to the class. We need to import `org.springframework.web.bind.annotation` package in our file, in order to implement it.

For more information [click here](#).

18) What is @RequestMapping annotation in Spring Boot?

The **@RequestMapping** annotation is used to provide routing information. It tells to the Spring that an HTTP request should map to the corresponding method. We need to import `org.springframework.web.annotation` package in our file.

For more information [click here](#).

19) How to create Spring Boot application using Spring Starter Project Wizard?

There is one more way to create Spring Boot project in STS (Spring Tool Suite). Creating project by usir IDE is always a convenient way. Follow the following steps in order to create a Spring Boot Application b using this wizard.

For more information [click here](#).

20) Spring Vs Spring Boot?

Spring is a web application framework based on Java. It provides tools and libraries to create a completely customized web application.

Whereas Spring Boot is a spring module which is used to create spring application project that can just run.



Javatpoint Services

JavaTpoint offers too many high quality services. Mail us on hr@javatpoint.com, to get more information about given services.

- Website Designing
- Website Development
- Java Development
- PHP Development
- WordPress
- Graphic Designing
- Logo
- Digital Marketing
- On Page and Off Page SEO
- PPC
- Content Development
- Corporate Training
- Classroom and Online Training
- Data Entry



Training For College Campus

JavaTPoint offers college campus training on Core Java, Advance Java, .Net, Android, Hadoop, PHP, Web Technology and Python. Please mail your requirement at hr@javatpoint.com.

Duration: 1 week to 2 week

Like/Subscribe us for latest updates or newsletter



LEARN TUTORIALS

- [Learn Java](#)
- [Learn Data Structures](#)
- [Learn C Programming](#)
- [Learn C++ Tutorial](#)
- [Learn C# Tutorial](#)
- [Learn PHP Tutorial](#)
- [Learn HTML Tutorial](#)
- [Learn JavaScript Tutorial](#)
- [Learn jQuery Tutorial](#)
- [Learn Spring Tutorial](#)

OUR WEBSITES

- [Javatpoint.com](#)
- [Hindi100.com](#)
- [Lyricsia.com](#)
- [Quoteperson.com](#)
- [Jobandplacement.com](#)

OUR SERVICES

- [Website Development](#)
- [Android Development](#)



College Campus Training

CONTACT

Address: G-13, 2nd Floor, Sec-3

Noida, UP, 201301, India

Contact No: 0120-4256464, 9990449935

[Contact Us](#)

[Subscribe Us](#)

[Privacy Policy](#)

[Sitemap](#)

[About Me](#)

© Copyright 2011-2018 www.javatpoint.com. All rights reserved. Developed by JavaTpoint.

Top 25 Most Frequently Asked Java Multithreading Interview Questions And Answers

In interviews we usually come across two topics [java collection interview questions](#) and multithreading interview questions. Multithreading is all about concurrency and threads. It is good to be prepared, so do not leave this topic at all. We are sharing important java multithreading interview questions and answers.

Q1. What is a thread?

Thread is a lightweight subprocess. Thread class belongs to `java.lang` package. Threads have their own stack. It's a way to take advantage of multiple CPU available in a machine. For example, if one thread takes 50 milliseconds to do a job, you can use 10 threads to reduce that task to 5 milliseconds.

Q2 What is the difference between Thread and Process?

A process can have many threads. Threads can execute any part of process. And same part of process can be executed by multiple threads.

Processes have their own address while Thread share the address space of the process that created it.

Thread has its own stack while in process all threads share a common system resource like heap memory.

Q3 What are the ways to implement Thread in java?

There are two ways to implement Thread in java.

1. By implementing Runnable interface in java and then creating Thread object from it.
2. By extending the Thread class.

Q4 What are the differences between implementing Runnable and extending Thread while creating a thread in java? Which one is better?

Implementing Runnable is better.

When we implement Runnable interface we can extend any other class as well but if we extends Thread class we can not extends any other class because java does not allow multiple inheritance.

You can find here the detailed answer of [difference between implementing Runnable and extends Thread](#).

Q5 What is a life cycle of a thread?

When we create a Thread instance in a java program, then its state is new. Then we start Thread, then it's state changes to Runnable(ready to run but not running yet).Execution of Threads depends upon ThreadScheduler. ThreadScheduler is responsible to allocate CPU to threads in Runnable thread pool and change their state to Running.Waiting,Blocked and Dead are the remaining states of the Thread.

So in short **new,runnable,running,waiting,blocked and dead** are the states a Thread can be in.

Q6 What is the difference between sleep and wait method in java?

Wait method releases the lock while sleep method doesn't release the lock.
Wait method belongs to java.lang.Object class while sleep method belongs to java.lang.Thread class.

You can find the detailed answer here [difference between wait and sleep method in java](#).

Q7 What is the difference between starting a thread with start() method and run() method?

This question is a bit tricky and might confuse you as well. The answer is when you call start() method, main method internally calls run() method to start newly created Thread, so run method is ultimately called by newly created Thread.

When you call run() method, its called in the same thread, no new thread is started which is the case when you call start() method.

Q8 What is the difference between user thread and daemon thread?

When we create a thread in java program, then it is called as user thread. We can not make a user thread to daemon thread if a thread is started.

The daemon threads are the low priority threads that provide the background support to the user threads. It provides services to the user threads. A child thread created from a daemon thread is also a daemon thread.

Q9 How to create a Daemon thread in java?

By setting the `setDaemon(true)` , we can create a daemon thread in java.

Q10 What is the significance of using volatile keyword?

When we make a variable volatile, then all the threads reads its value directly from the memory and don't cache it. This make sure the shared variables are consistently updated.

volatile is a keyword that can only be used with variables.

Q11 Is it possible to start a thread twice?

No, there is no possibility to start a thread twice. If we do so , then it will throw an Exception.

Q12 What is synchronization?

Synchronization is the capability to control the access of multiple threads to any shared resource.

The main advantage of synchronization is

- a. to avoid consistency problem
- b. to avoid thread interference

Q13 Which is more preferred - synchronization block or synchronization method?

Synchronized block is the more preferred way because it doesn't lock the object while synchronized methods lock the object. Synchronized method will stop multiple synchronized blocks in the class, even though they are not related, from the execution and put them in the wait state to get the lock on the object.

Q14 Difference and similarities between sleep and yield method?

Sleep method throws the interrupted exception if another thread interrupts the sleeping thread while yield method does not throw the interrupted exception.

Thread.sleep() method does not cause currently executing thread to give up monitors while yield method gives up the monitor.

You can find detailed explanation of [difference between sleep and yield method in java](#).

Q15 What is deadlock?

Deadlock is a situation where two threads are waiting for each other to release locks held by them on resources. For example

Thread 1 : locks resource A, waits for resource B

Thread 2 : locks resource B, waits for resource A

Q16 Write a program to create a Deadlock in java?

You can find the answer here [Program to create a deadlock in java.](#)

Q17 What measures you should take to avoid deadlock?

1. Lock specific member variables of the class rather than locking whole class.
2. Use join() method, if possible try to use join method ,although it may refrain us from taking full advantage of multithreading environment because threads will start and end sequentially but it can be handy in avoiding deadlocks.
3. If possible try to avoid nested synchronization blocks.

Q18 What do you understand by Thread priority?

Every thread has a priority. Its value is *int* which ranges from 1 to 10 where 1 being the lowest priority and 10 being the highest priority.

Usually higher priority threads get higher precedence in execution but it depends on the ThreadScheduler implementation which is OS dependent.

We can specify the priority of thread but it does not guarantee that higher priority thread will get executed before the lower priority thread.

Q19 What is the difference between class lock and object lock?

Threads can acquire object lock by entering synchronized methods. Threads can acquire lock on class's class object by entering the static synchronized methods.

Multiple objects of class may exist and every Object has its own lock. In class lock multiple objects of class may exist but there is always one class's class object lock available.

Q20 What is the difference between Callable and Runnable?

Callable throws checked exception while Runnable does not throw checked exception.

Return type of Runnable is void that is it does not return any value while Callable can return a Future object.

You can find the detailed explanation of [difference between callable and runnable](#).

Q21 What is the difference between time slicing and preemptive scheduling?

In preemptive scheduling the higher priority task executes until it enters the waiting or dead states or higher priority task comes into existence. In time slicing, a task runs for a predefined slice of time and then reenters the pool of ready tasks.

Q22 Can a constructor be synchronized?

No, Constructor can not be synchronized.

Q23 What is race condition in java and how we can solve it?

When more than one thread try to access same resource without synchronization causes race condition.

We can solve race condition by using a synchronized block or synchronized method.

Q24 How threads communicate with each other?

Threads can communicate with each other using wait(), notify(), notifyAll() methods.

Q25 Why wait(), notify() and notifyAll() method have to be called from the synchronized context?

When a Thread calls wait() on any Object, it must have the monitor on Object that it will leave and goes in wait state until any other Thread call notify() on this Object. Similarly when a thread calls notify() on any Object, it leaves the monitor on the Object and other waiting threads can get the monitor on the Object. Since all these threads require Thread to have Object monitor, that can be achieved only by synchronization. That is why wait(), notify() and notifyAll() method have to be called from the synchronized context.

Q26 What is ThreadLocal variable in java?

ThreadLocal can be used to create ThreadLocal variables. We know that all threads of an Object shares its variables. So if the variable is not thread safe then we can use synchronization. If we want to avoid synchronization then we can use ThreadLocal variable. Each thread has its own ThreadLocal variable and they can use its get() and set() method to get the default value or change its value local to Thread.

Q27 What is Threadpool?

Threadpool manages the pool of worker threads. There is a queue in which the tasks are kept waiting for execution.

Q28 Can you find whether thread holds lock() on an object or not?

holdsLock(Object) method can be used to determine whether current thread holds the lock on monitor of specified object.

The method holdsLock(Object) returns true if the thread holds lock or monitor of the specified object.

If you face any java multithreading question which is not in above questions then please mention in the comments.

About The Author

Subham Mittal has worked in Oracle for 3 years .

For more java articles ,Click here to [Subscribe](#)

[JavaHungry](#)

You might also like:



[Top 15 Frequently Asked HashMap Interview Questions in Java](#)



[Top 50 Java Array Interview Questions and Answers](#)

[Top 50 Java String Interview Questions and Answers](#)

[Top 28 Most Frequently Asked Interview Core Java Interview Questions And Answers 2018](#)

[6 Difference between HashMap and HashTable : Popular Interview Question in Java with Example](#)

[Top 40 Java Web Services Interview Questions and Answers](#)

[Core Java Coding / Programming Questions and Answers : Technical Interview in Java](#)



Login



Leave your comment...

Comments

Sort by Best

Be the first to comment!



ADD WIDGETPACK TO YOUR SITE

POWERED BY **WIDGET PACK™**

[Return to Top ↑](#) Affiliate-Disclaimer Privacy Java Hungry Copyright © 2021 · All Rights Reserved. Designed by studiopress