
Top Java Interview Questions And Answers

Here are top 53 objective type sample Java Interview questions and their answers are given just below to them. These sample questions are framed by experts from Intellipaat who trains for Java training (<https://intellipaat.com/java-training/>) to give you an idea of type of questions which may be asked in interview. We have taken full care to give correct answers for all the questions. Do comment your thoughts Happy Job Hunting!

Top Answers to Java Interview Questions

1. What do you understand by Java?

- Java is an object-oriented computer language.
- It is a high-level programming language developed by James Gosling in Sun Microsystem in 1995.
- Java is a fast, secure and reliable language used for many games, devices and applications.

Go through this Java Tutorial (<https://intellipaat.com/tutorial/java-tutorial/introduction-java/>) to get a better understanding of the concept.

2. Compare Java & Python

Criteria	Java	Python
Ease of use	Good	Very Good
Speed of coding	Average	Excellent
Data types	Static	Dynamically

	typed	typed
Data Science & machine learning applications	Average	Very Good

3. Outline the major Java features.

- Object-Oriented – java is based on object-oriented programming where the class and methods describe about the state and behavior of object.
- Portable – Java program gets converted into Java Byte Codes that can be executed on any platform without any dependency.
- Platform independent – java works on “write once and run anywhere” as it supports multiple platforms like Windows, Linux, Mac, Sun Solaris, etc.
- Robust – Java has a strong memory management as there is no pointer allocations. It has automatic garbage collection that prohibits memory leaks.
- Interpreted – java compiler converts the codes into Java Byte Codes which are then interpreted and executed by Java Interpreter.

Become Master of Java by going through this online Java training course (<https://intellipaat.com/java-training/#about-course>).

4. What do you mean by Object?

An object consists of methods and class which depict its state and perform operations. A java program contains a lot of objects instructing each other their jobs. This concept is a part of core java.

Get a clear understanding of Java through this video

[https://www.youtube.com/watch?v=...](#)

(<https://intellipaat.com/java-training/#course-preview>).

5. What is class in Java?

Java encapsulates the codes in various classes which define new data types. These new data types are used to create objects.

6. Differentiate between JDK, JRE and JVM.

- JVM stands for Java Virtual Machine which provides runtime environment for Java Byte Codes to be executed.
- JRE (Java Runtime Environment) that includes sets of files required by JVM during runtime.
- JDK (Java Development Kit) consists of JRE along with the development tools required to write and execute a program.

7. Define Inheritance.

Java includes the feature of inheritance which an object-oriented programming concept. Inheritance lets a derived class to inherit the methods of a base class.

Read this tutorial (<https://intellipaat.com/tutorial/java-tutorial/inheritance-is-a/>) to get clear understanding of Inheritance in Java.

8. Explain method overloading.

When a Java program contains more than one methods with the same name but different properties, then it is called method overloading.

same method overriding.

9. Compare Overloading and Overriding.

Overloading refers to the case of having two methods of same name but different properties, but overriding occurs when there are two methods of same name and properties, but one is in child class and one is in parent class.

10. Explain the creation of a thread-safe singleton in Java using double-checks locking

Singleton is created with double checked locking as before Java 5 acts as an broker and it's been possible to have multiple instances of Singleton when multiple threads creates an instance of Singleton at the same time. Java 5 made it easy to create thread-safe Singleton using Enum. Using a volatile variable is essential for the same.

Learn more about Java threading in this tutorial.
(<https://intellipaat.com/tutorial/java-tutorial/java-multithreading/>)

11. Differentiate between StringBuffer and StringBuilder in Java programming.

String Buffer	String Builder
StringBuffer methods are synchronized	StringBuilder is non synchronized
Storage area is Heap and modified easily.	Storage is Heap and can be modified.

StringBuffer is thread safe.	StringBuilder is fast as it is not thread safe
Performance is very slow	Performance is very fast.

Go through this Java tutorial (<https://intellipaat.com/tutorial/java-tutorial/java-strings/>) to get better understanding of String.

12. Difference between Array list And Vector.

Array List	Vector
Array List is not synchronized.	Vector is synchronized.
Array List is fast as it's non-synchronized.	Vector is slow as it is thread safe.
If an element is inserted into the Array List, it increases its Array size by 50%.	Vector defaults to doubling size of its array.
Array List does not define the increment size.	Vector defines the increment size.
Array List can only use Iterator for traversing an Array List.	Except Hashtable, Vector is the only other class which uses both Enumeration and Iterator.

Learn all about Array in this comprehensive Java tutorial (<https://intellipaat.com/tutorial/java-tutorial/java-array/>).

13. Differentiate between Iterator and Enumeration?

Iterator	Enumeration
Iterator is an interface found in the java.util package.	Enumeration is an object that generates elements one at a time. Used for passing through a collection for unknown size.
Uses 3 methods to interface such as: 1. i) hasNext() 2. ii) next() 3. iii) remove()	Methods used are: 1. i) hasMoreElements() 2. ii) nextElement()
Iterators allow removing elements from the given collection during the iteration with well-defined semantics.	It is used for passing through a collection, usually of unknown size.
Iterator method names have been improved.	The traversing of elements can only be done once per creation

14. Difference between the Inner Class and Sub Class.

Inner Class	Sub Class
It's a class which is nested within another class.	It's a class which inherits from another class called super class.
Inner class provides the access rights for the class which is nesting it and that can access all variables and methods defined in the outer class.	Sub-class provides access to all public and protected methods and fields of its super class.

15. Can we execute any code, even before the main method? Explain?

Yes, We can execute any code, even before the main method. We are using a static block of code in the class when creating the objects at load time of class. Any statements within this static block of code will get executed one time while loading the class, even before the creation of objects in the main method.

Prepare yourself for the Java Certification with comprehensive Online Java Training (<https://intellipaat.com/java-training/#about-course>).

16. How can we restrict inheritance for a class?

We can restrict inheritance for class by following steps.

1. By using final keyword
2. If we make all method final, then we cannot override that.
3. By using private constructors
4. By using Javadoc comment (//)

17. Java doesn't support multiple inheritance. Why?

Java doesn't support multiple inheritance. Because we cannot use different methods in one class it creates an ambiguity.

Example:

```
class Intellipaat1
{
void test()
{
```

```

system.out.println("test() method");
}
}class Intellipaat2
{
void test()
{
system.out.println("test() method");
}
}Multiple inheritance
class C extends Intellipaat1, Intellipaat2
{
.....
.....
}

```

Intellipaat1 and Intellipaat2 test() methods are inheriting to class C

So which test() method C class will take. As Intellipaat1 & Intellipaat2 class test () methods are different, So here we would face ambiguity.

18. Are constructors inherited? Can a subclass call the parent's class constructor?

We cannot inherit a constructor. We create an instance of a subclass using a constructor of one of its superclass. Because override the superclass constructor is not our wish so that, we override a superclass constructor, then we destroy the encapsulation abilities of the language.

Check the insightful tutorial to learn more about Java Constructors (<https://intellipaat.com/tutorial/java-tutorial/java-constructors/>).

19. Define JSON?

Expansion of JSON is “JavaScript Object Notation”, It is a much lighter and readable alternative to XML. It is an independent and easily parse-able in all programming

languages. It is primarily used for Communicating between client – server or server -server communication. It is a much lighter and readable alternative to XML.

20. What are the advantages of JSON over XML?

The advantages of JSON over XML are

1. JSON is Lighter and faster than XML
2. Better understandable.
3. Easy to parse and conversion to objects for information consumption.
4. Support multiple data types – JSON support string, number, array, Boolean but XML data are all strings.

21. Name the methods of Object Class?

clone() – This method helps to create and return a copy of the object.

equals() – This method helps to compare

finalize() – Called by the garbage collector on an object

getClass() – It helps to return the runtime class of an object.

hashCode() – Helps to return a hash code value for the object.

toString() – helps to return a string representation of the object.

notify(), notifyAll(), and wait() – It helps to synchronize the activities of independently running threads in a program.

Read this tutorial (<https://intellipaat.com/tutorial/java-tutorial/java-methods/>) to learn more about Java Methods.

22. Define content negotiation?

If we have visited website to searching the information, we will get the information in different languages and in different formats when a client makes an HTTP request to a server, the client can also specify the media types here. The client can specify what it can accept back from the host and on the basis of availability the host will return to the client. This is known as content negotiation because client and server negotiated on the language and format of the content to be shared.

23. Can we import same package/class two times? Will the JVM load the package twice at runtime?

A package or class can be inherited multiple times in a program code. JVM and compiler will not create any issue. Moreover JVM automatically loads the class internally once regardless of times it is called in the program.

24. Define Abstract class?

A class which contains the abstract keyword in its declaration is known as abstract class. It can have abstract and non-abstract methods (method with body).

1. This class can have public, private, protected or constants and default variables.
2. It needs to be extended and its method implemented. It cannot be instantiated.
3. If a class has at least one abstract method, then the class must be declared abstract.

25. Describe the Annotations.

- Java Annotations is a tag which symbolizes metadata associated with class, interface, methods, fields, etc.
- Annotations do not directly influence the operations.
- The additional information carried by annotations are utilized by java compiler and JVM.

26. Java doesn't use pointers. Why?

Pointers are susceptible and slight carelessness in their use may result in memory problems and hence Java basically manages their use.

27. Distinguish between static loading and dynamic class loading?

Static loading – Classes are loaded statically with operator “new”. Dynamic class loading – It is a technique for programmatically invoking the functions of a class loader at run time. The syntax is `Class.forName (Test className);`

28. Explain Struts 1 Classes are not Thread Safe whereas Struts 2 classes are thread safe?

Struts 1 actions are a singleton. So all threads operate on the single action object and hence make it thread-unsafe. Struts 2 actions are not a singleton and a new action object copy are created each time a new action request is made and hence it threads safe.

29. Define JAXP & JAXB?

JAXP: – Stands for Java API for XML Processing. This provides a common interface for creating and using DOM, SAX, and XSLT APIs in Java regardless of which vendor's implementation is actually being used.

JAXB: – It Stands for Java API for XML Binding. This standard defines a system for a script out Java objects as XML and for creating Java objects from XML structures.

30. Define an enumeration?

Usually, we called enumeration as an enum. An enumeration is an interface containing methods for accessing the original data structure from which the enumeration is obtained. It allows sequential access to all the elements stored in the collection.

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

In Java, there is no way to find out the actual size of an object on the heap.

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

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

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

Java.Lang.throwable – It is the super class of all exception classes and all exception classes are derived from this base class.

34. Why we used Vector class?

The Vector class provides the ability to execute a growable array of objects. Vector proves to be very useful if you don't know the size of the array in advance, or we need one that can change sizes over the lifetime of a program.

35. What is the difference between transient and volatile variable in Java?

Transient: In Java, it is used to specify the variable is not being serialized. Serialization is a process of saving an object's state in Java. When we want to persist and the object's state by default, all instance variables in the object are stored. In some cases, if we want to avoid persisting few variables because we don't have the necessity to transfer across the network. So, declare those variables as transient.

If the variable is confirmed as transient, then it will not be persisted. Transient keyword is used with that instance variable which will not participate in the serialization process. We cannot use static with a transient variable as they are part of the instance variable. Volatile: Volatile keyword is used with only variable in Java and it guarantees that the value of the volatile variable will always be read from main memory and not from Thread's local cache. it can be static

local cache, it can be static.

36. Why does Map interface not extend the Collection interface in the Java Collections Framework?

The map interface is not compatible with the Collection interface. Because Map requires key as well as value, for example: if we want to add key-value pair so we will use `put(Object key, Object value)`. There are two parameters required to add an element to HashMap object. In Collection interface `add(Object o)` has only one parameter.

The other reasons are Map supports `valueSet`, `keySet` as well as other suitable methods which have just different views from the Collection interface.

37. Mention the uses of Synchronized block

We are using Synchronized block because

- It helps to lock an object for every shared resource.
- The scope of synchronized block is smaller than the method.

38. What are the functions of hashCode() method?

This method returns a hash code value (an integer number).

And also returns the same integer number, if two keys (by calling `equals()` method) are same.

But in sometimes two hash code numbers can have different or same keys.

39. What is the default size of load factor in hashing based collection?

Default Size = 0.75. The default capacity is computed as initial capacity * load factor.

40. What are the differences between the JSP custom tags & Java beans?

- Custom tags can manipulate JSP content, but beans cannot.
- Composite operations can be reduced to a considerably simpler form with custom tags than with beans.
- Custom tags require reasonably a bit more work to set up than do beans.
- Custom tags are available only in JSP 1.1 and so on, but beans can be used in all JSP 1.x versions

41. Explain Expression Language in JSP?

The expression language is used in JSP to simplify the accessibility of objects and It provides many objects that can be used directly like param, requestScope, sessionScope, applicationScope, request, session etc.

42. What are implicit objects?

Implicit objects, also called as pre-defined variables, are created by JSP Engine inside the service method so that it can be accessed directly without being declared explicitly.

43. Define a cookie and what are the difference between session and cookie?

A cookie is a small piece of information, this piece of information sends to the browser by a web server. The browser stores the cookies for every Web server in a local file. In a future request or future reference, the browser sends all stored cookies for that detailed Web server.

The difference between cookies and sessions are:

- The session is working, regardless of the settings on the client browser. The client may have selected to disable cookies.
- The session and cookies also differ in storing the amount of information. The HTTP session is able to store any Java object, while a cookie can only store String objects.

44. What is HTTP Tunneling?

Tunneling refers to encapsulating private network data/information and transmitting it through a public network. Instead of sending data as a packet, tunneling encrypts the data and encapsulates in a connection. It allows the outside clients to collect all the data sent by client-side Object Request Broker (ORB) that needs to be sent to server-side ORB. HTTP tunneling masks other protocol requests as HTTP requests.

45. What is a function of IOC container in spring?

IOC container is responsible to:

- create the instance

- configure the instance, and
- assemble the dependencies

46. What is lazy loading in hibernate?

Lazy loading is a kind of setting that decides whether to load the child entities with the parent entities or not. When enabling this feature the associated entities will be loaded only when it is requested directly. The default value of this setting is true which stops child entities from loading.

47. How can we fetch records by spring JdbcTemplate?

We can fetch records from the database by the query method of JdbcTemplate. There are two interfaces to do this:

1. ResultSetExtractor
2. RowMapper

48. What is the front controller class of Spring MVC?

The Dispatcher Servlet class works as the front controller in Spring MVC.

49. What are the states of an object in hibernate?

The states of an object in hibernate are-

1. Transient: When objects are just created having no primary key are in transient state. Here the objects are associated with any session.

2. Persistent: When the session of an object is just opened and its instance is just saved or retrieved, it is said to be in persistent state.
3. Detached: When the session of an object is closed, it is said to be in detached state.

50. How to make an immutable class in hibernate?

If we mark a class as mutable="false", the class will be treated as an immutable class. The default value of mutable is "true".

51. What is hash-collision in Hashtable? How was it handled in Java?

In Hashtable , if two different keys have the same hash value then it leads to hash -collision. A bucket of type linked list used to hold the different keys of same hash value.

52. Write a syntax to convert a given Collection to SynchronizedCollection ?

`Collections.synchronizedCollection(Collection collectionObj)` will convert a given collection to synchronized collection.

53. Write a code to make Collections readOnly?

We can make the Collection readOnly by using the following lines code:

```
General : Collections.unmodifiableCollection(Collection c)Col
```

```
lections.unmodifiableMap(Map m)
Collections.unmodifiableList(List l)
Collections.unmodifiableSet(Set s)
```

54. What is meant by binding in RMI?

Binding is the process of associating or registering a name for a remote object, which can be used as a further, in order to look up that remote object. A remote object can be associated with a name using the bind / rebind methods of the Naming class.

55. What are latest features introduced with Java 8?

The below latest features are introduced in Java 8. Lambda Expressions, Interface Default and Static Methods, Method Reference, Parameters Name, Optional Streams, Concurrency.

56. Name few Java 8 annotations?

1. @Functional Interface annotation

2. @Repeatable annotation,

@Functional Interface annotation: It was introduced in Java SE 8, indicates that the type declaration is intended to be a functional interface, as defined by the Java Language Specification. @Repeatable annotation: introduced in Java SE 8, indicates that the marked annotation can be applied many times to the same declaration or type use.

57. Distinguish between a predicate and a function?

A predicate takes one argument and returns a Boolean value.

A function takes one argument and returns an object.

Both are useful for evaluating lambda expressions.

58. Write a code to sort a list of strings using Java 8 lambda expression?

```
. private void sortUsingJava8(List names){  
Collections.sort(names, (p1, p2) -> p1.compareTo(p2));  
}
```

59. What is Nashorn in Java8?

With Java 8, Nashorn, a much-improved javascript engine is introduced and Nashorn replaced the existing Rhino. It provides 2 to 10 times better performance, as it directly compiles the code in memory and passes the bytecode to JVM. Nashorn uses invoke the dynamic feature, introduced in Java 7 to improve performance.

60. Define a StringJoiner and write a sample code?

StringJoiner is an util method to construct a string with the desired delimiter.

```
Sample CodeStringJoiner strJoiner = new StringJoiner(".");  
strJoiner.add("AAA").add("BBB");  
System.out.println(strJoiner);  
OutPut:  
AAA.BBB
```

61. Can you execute javascript code from Java 8
code here?

code base:

Yes! We can execute javascript code from Java8 code based by using ScriptEngineManager, We called JavaScript code and interpreted in Java.

This blog will help you get a better understanding of Mobile Technology for Android and iOS using Java (<https://intellipaat.com/blog/cross-platform-mobile-technology-for-android-and-ios-using-java/>) !

« Previous (<https://intellipaat.com/interview-question/data-warehouse-interview-questions/>)

Next » (<https://intellipaat.com/interview-question/microstrategy-interview-questions/>)