

1) Difference between String, StringBuffer and StringBuilder in Java?

String is immutable while both StringBuffer and StringBuilder is mutable, which means any change e.g. converting String to upper case or trimming white space will produce another instance rather than changing the same instance. On later two, StringBuffer is synchronized while StringBuilder is not, in fact its a ditto replacement of StringBuffer added in Java 1.5.

2) Difference between extends Thread vs implements Runnable in Java?

Difference comes from the fact that you can only extend one class in Java, which means if you extend Thread class you lose your opportunity to extend another class, on the other hand if you implement Runnable, you can still extend another class.

3) Difference between Runnable and Callable interface in Java?

Runnable was the only way to implement a task in Java which can be executed in parallel before JDK 1.5 adds Callable. Just like Runnable, Callable also defines a single call() method but unlike run() it can return values and throw exceptions.

4) Difference between ArrayList and LinkedList in Java?

In short, ArrayList is backed by array in Java, while LinkedList is just collection of nodes, similar to linked list data structure. ArrayList also provides random search if you know the index, while LinkedList only allows sequential search. On other hand, adding and removing element from middle is efficient in LinkedList as compared to ArrayList because it only require to modify links and no other element is rearranged.

5) What is difference between wait and notify in Java?

Both wait and notify methods are used for inter thread communication, where wait is used to pause the thread on a condition and notify is used to send notification to waiting threads. Both must be called from synchronized context e.g. synchronized method or block.

6) Difference between HashMap and Hashtable in Java?

Though both HashMap and Hashtable are based upon hash table data structure, there are subtle difference between them. HashMap is non synchronized while Hashtable is synchronized and because of that HashMap is faster than Hashtable, as there is no cost of synchronization associated with it. One more minor difference is that HashMap allows a null key but Hashtable doesn't.

7) Difference between TreeSet and TreeMap in Java?

Though both are sorted collection, TreeSet is essentially a Set data structure which doesn't allow duplicate and TreeMap is an implementation of Map interface. In reality, TreeSet is implemented via a TreeMap, much like how HashSet is implemented using HashMap

8) Write a Java program to print Fibonacci series? ([solution](#))

Fibonacci series is a series of number on which a number is equal to sum of previous two numbers i.e. $f(n) = f(n-1) + f(n-2)$. This program is used to teach recursion to students but you can also solve it without recursion. Check out the solution for both iterative and recursive solution of this problem. In telephonic interview, this question is not that common but sometime interviewer also wants to check your problem solving skill using such questions.

9) Write a Java program to check if a number is Prime or not? ([solution](#))

A number is said prime if it is not divisible by any other number except itself. 1 is not considered prime, so your check must start with 2. Simplest solution of this to check every number until the number itself to see if its divisible or not. When Interviewer will ask you to improve, you can say that checking until square root of the number. If you can further improve the algorithm, you will more impress your interviewer. check out the solution for how to do this in Java

10) How to Swap two numbers without using temp variable? ([solution](#))

This question is ages old. I have first seen this question way back in 2005 but I am sure its even older than that. Good thing about this problem is that except XOR trick all solution has some flaws, which is used to test whether candidate really knows his stuff or not. Check out the solution for all three possible solution and drawback of each.

11) How to check if linked list contains loop in Java? ([solution](#))

This is another problem solving question which is very popular in telephonic and screening round. This is a great question to test problem solving skill of candidate, especially if he has not seen this question before. It also has a nice little followup to find the starting of the loop. See the solution for a Java program which finds loop in singly linked.

12) Write Java program to reverse String without using API? ([solution](#))

One more question to test problem solving skill of candidate. You wouldn't expect these kind of question in telephonic round of Java interview but these questions have now become norms. All interviewer is looking it for logic, you don't need to write the code but you should be able to think of solution.

13) Difference between Serializable and Externalizable in Java? ([detailed answer](#))

Serializable is a marker interface with no methods defined it but Externalizable interface has two methods defined on it e.g. readExternal() and writeExternal() which allows you to control the serialization process. Serializable uses default serialization process which can be very slow for some application.

14) Difference between transient and volatile in Java? ([detailed answer](#))

transient keyword is used in Serialization while volatile is used in multi-threading. If you want to exclude a variable from serialization process then mark that variable transient. Similar to static variable, transient variables are not serialized. On the other hand, volatile variables are signal to compiler that multiple threads are interested on this variable and it shouldn't reorder its access. volatile variable also follows happens-before relationship, which means any write happens before any read in volatile variable. You can also make non atomic access of double and long variable atomic using volatile.

15) Difference between abstract class and interface? ([detailed answer](#))

From Java 8 onwards difference between abstract class and interface in Java has minimized, now even interface can have implementation in terms of default and static method. BTW, in Java you can still extend just one class but can extend multiple inheritance. Abstract class is used to provide default implementation with just something left to customize, while interface is used heavily in API to define contract of a class.

16) Difference between Association, Composition and Aggregation? ([detailed answer](#))

Between association, composition and aggregation, composition is strongest. If part can exists without whole than relationship between two class is known as aggregation but if part cannot exists without whole than relationship between two class is known as composition. Between Inheritance and composition, later provides more flexible design.

17) What is difference between FileInputStream and FileReader in Java? ([detailed answer](#))

Main difference between FileInputStream and FileReader is that former is used to read binary data while later is used to read text data, which means later also consider character encoding while converting bytes to text in Java.

18) How do you convert bytes to character in Java? ([detailed answer](#))

Bytes are converted to character or text data using character encoding. When you read binary data from a file or network endpoint, you provide a character encoding to convert those bytes to equivalent character. Incorrect choice of character encoding may alter meaning of message by interpreting it differently.

19) Can we have return statement in finally clause? What will happen? ([detailed answer](#))

Yes, you can use return statement in finally block, but it will not prevent finally block from being executed. BTW, if you also used return statement in try block then return value from finally block will override whatever is returned from try block.

20) Can you override static method in Java? ([detailed answer](#))

No, you cannot override static method in Java because they are resolved at compile time rather than runtime. Though you can declare and define static method of same name and signature in child class, this will hide the static method from parent class, that's why it is also known as method hiding in Java.

21) Difference between private, public, package and protected in Java? ([detailed answer](#))

All four are access modifier in Java but only private, public and protected are modifier keyword. There is no keyword for package access, its default in Java. Which means if you don't specify any access modifier than by default that will be accessible inside the same package. Private variables are only accessible in the class they are declared, protected are accessible inside all classes in same package but only on sub class outside package and public variables e.g. method, class or variables are accessible anywhere. This is highest level of access modifier and provide lowest form of encapsulation.

22) 5 Coding best practices you learned in Java? ([detailed answer](#))

If you are developing on a programming language for couple of years, you sure knows lots of best practices, by asking couple of them, Interviewer just checking that you know your trade well. Here are my 5 Java best practices :

- Always name your thread, this will help immensely in debugging.
- Use StringBuilder for string concatenation
- Always specify size of Collection, this will save lot of time spent on resizing
- Always declare variable private and final unless you have good reason.
- Always code on interfaces instead of implementation
- Provide dependency to method instead they get it by themselves, this will make your code unit testable.

23) Write a Program to find maximum and minimum number in array? ([solution](#))

This is another coding question test problem solving ability of candidate. Be ready for couple of follow up as well depending upon how you answer this question. Simplest way which comes in mind is to sort the array and then pick the top and bottom element. For a better answer see the solution.

24) Write a program to reverse Array in place? ([solution](#))

Another problem solving question for Java programmers. Key point here is that you need to reverse the array in place, which means you cannot use additional buffer, one or two variable will be fine. Note you cannot use any library code, you need to create your own logic.

25) Write a program to reverse a number in Java? ([solution](#))

This is an interesting program for a very junior programmer, right from the college but can sometime puzzle developer with couple of years of experience as well. Most of these developer does very little coding so they found these kind of questions challenging. Here the trick is to get the last digit of the number by using modulus operator (%) and reducing number in each go by using division operator (/). See the solution for how to do that in Java.

26) Write a Program to calculate factorial in Java? ([solution](#))

Another beginners coding problem, good for telephonic interview because you can differentiate a guy who can write program to the guy who can't. It's also good to see if developer is familiar with both recursive and iterative algorithm and pros and cons of each. You can also ask lots of follow up e.g. how to improve performance of algorithm?

Since factorial of a number is equal to number multiplied by factorial of previous number, you can cache those value instead of recalculating them, this will impress your interviewer a bit. See the solution for full code example.

27) What is difference between calling start() and run() method of Thread? ([detailed answer](#))

You might have heard this question before, if calling start() method calls the run() method eventually then why not just call the run() method? Well the difference is, start method also starts a new thread. If you call the run method directly then it will run on same thread not on different thread, which is what original intention would be.

28) Write a Program to solve Producer Consumer problem in Java? ([solution](#))

A very good question to check if candidate can write inter thread communication code or not. If a guy can write producer consumer solution by hand and point out critical section and how to protect, how to communicate with thread then he is good enough to write and maintain your concurrent Java program. This is the very minimum requirement for a Java developer and that's why I love this question, it also has several solution e.g. by using concurrent collections like blocking queue, by using wait and notify and by using other synchronizers of Java 5 e.g. semaphores.

29) How to find middle element of linked list in one pass? ([solution](#))

Another simple problem solving question for warm up. In a singly linked list you can only traverse in one direction and if you don't know the size then you cannot write a loop to exactly find out middle element, that is the crux of the problem. One solution is by using two pointers, fast and slow. Slower pointer moves 1 step when faster pointer move to 2 steps, causing slow to point to middle when fast is pointing to end of the list i.e. null. Check out solution for Java code sample.

30) What is equals() and hashCode() contract in Java? Where does it used? ([detailed answer](#))

One of the must ask question in Java telephonic interview. If a guy doesn't know about equals() and hashCode() then he is probably not worth pursuing further because its the core of the Java fundamentals. The key point of contract is that if two objects are equal by equals() method then they must have same hashCode, but unequal object can also have same hashCode, which is the cause of collision on hash table based collection e.g HashMap. When you override equals() you must remember to override hashCode() method to keep the contract valid.

31) Why wait and notify methods are declared in Object class? ([detailed answer](#))

This question is more to find out how much experience you really have and what is your thinking towards Java API and its design decision. Similar question is why String is immutable in Java? Well, true answer can only be given by Java designers but you can reason something. For example, wait and notify methods are associated with locks which is owned by object not thread, and that's why it make sense to keep those method on java.lang.Object class. See the detailed answer for more discussion and reasoning.

32) How does HashSet works in Java? ([detailed answer](#))

HashSet is internally implemented using HashMap in Java and this is what your interviewer wants to hear. He could then quiz you with some common sense based question e.g. how can you use HashMap because its needs two object key and value? what is the value in case of HashSet? Well, in case of HashSet a dummy object is used as value and key objects are the actual element on Set point of view. Since HashMap doesn't allow duplicate key it also follows contract of set data structure to not allow duplicates. See detailed answer for more analysis and explanation.

33) What is difference between synchronize and concurrent Collection in Java? ([detailed answer](#))

There was time, before Java 1.5 when you only have synchronized collection if you need them in a multi-threaded Java program. Those classes were plagued with several issue most importantly performance because they lock the whole collection or map whenever a thread reads or writes. To address those issue, Java released couple of Concurrent collection classes e.g. ConcurrentHashMap, CopyOnWriteArrayList and BlockingQueue to provide more scalability and performance.

34) What is difference between Iterator and Enumeration in Java? ([detailed answer](#))

Main difference is that Iterator was introduced in place of Enumeration. It also allows you to remove elements from collection while traversing which was not possible with Enumeration. The methods of Iterator e.g. hasNext() and next() are also more concise then corresponding methods in Enumeration e.g. hasMoreElements(). You should always use Iterator in your Java code as Enumeration may get deprecated and removed in future Java release.

35) What is difference between Overloading and Overriding in Java? ([detailed answer](#))

Another frequently asked question from telephonic round of Java interviews. Though both overloading and overriding are related with methods of same names but they have different characteristics e.g. overloaded methods must have different method signature than original method but overridden method must have same signature. Also, overloaded methods are resolved at compile time while overridden methods are resolved at runtime. See the detailed answer for more analysis and differences.

36) Difference between static and dynamic binding in Java? ([detailed answer](#))

This is usually asked as follow-up of previous question, static binding is related to overloaded method and dynamic binding is related to overridden method. Method like private, final and static are resolved using static binding at compile time but virtual methods which can be overridden are resolved using dynamic binding at runtime.

37) Difference between Comparator and Comparable in Java? ([detailed answer](#))

This is one more basic concept, I expect every Java candidate to know. You will deal with them on every Java project. Several core classes in Java e.g. String, Integer implements Comparable to define their natural sorting order and if you define a value class or a domain object then you should also implement Comparable and define natural ordering of your object. Main difference between these two is that, you could create multiple Comparator to define multiple sorting order based upon different attribute of object. Also, In order to implement Comparable you must have access of the class or code, but you can use Comparator without having source code of class, all you need is the JAR file of particular object. That's why Comparator is very powerful to implement custom sorting order and from Java 8 you can do it even more elegantly, as seen here.

38) How do you sort ArrayList in descending order? ([solution](#))

You can use Collections.sort() method with reverse Comparator, which can sort elements in the reverse order of their natural order e.g.

```
List<String> listOfString = Arrays.asList("London", "Tokyo", "NewYork");  
Collections.sort(listOfString, Collections.reverseOrder());  
System.out.println(listOfString); //[Tokyo, NewYork, London]
```


39) What is the difference between PATH and CLASSPATH in Java?

PATH is an environment variable which points to Java binary which is used to run Java programs. CLASSPATH is another environment variable which points to Java class files or JAR files. If a class is not found in CLASSPATH then Java throws ClassNotFoundException.

40) What is the difference between Checked and Unchecked Exception in Java? ([detailed answer](#))

Checked exception ensures that handling of the exception is provided and its verified by compiler also, while for throwing unchecked exception no special provision is needed e.g. throws clause. A method can throw unchecked exceptions without any throw clause