ALL THE BEST DEVELOPERS

Java Basics

1. What is the difference between JDK and JRE?

2. What is Java Virtual Machine (JVM)?

3. What are the different types of memory areas allocated by JVM?

4. What is JIT compiler?

5. How Java platform is different from other platforms? 6. Why people say that Java is 'write once and run anywhere' language? 7. How does ClassLoader work in Java?

8. Do you think 'main' used for main method is a keyword in Java?

9. Can we write main method as public void static instead of public static void?

10. In Java, if we do not specify any value for local variables, then what will be the default value of the local variables?

11. Let say, we run a java class without passing any arguments. What will be the value of String array of arguments in Main method?

12. What is the difference between byte and char data types in Java? OOPS

13. What are the main principles of Object Oriented Programming?

14. What is the difference between Object Oriented Programming language and Object Based Programming language?

15. In Java what is the default value of an object reference defined as an instance variable in an Object?

16. Why do we need constructor in Java?

17. Why do we need default constructor in Java

classes? 18. What is the value returned by Constructor

in Java? 19. Can we inherit a Constructor?

20. Why constructors cannot be final, static, or abstract in Java? Inheritance

21. What is the purpose of 'this' keyword in java?

22. Explain the concept of Inheritance?

23. Which class in Java is superclass of every other

class? 24. Why Java does not support multiple

inheritance? 25. In OOPS, what is meant by

composition?

26. How aggregation and composition are different

concepts? 27. Why there are no pointers in Java?

28. If there are no pointers in Java, then why do we get NullPointerException?

29. What is the purpose of 'super' keyword in java? 30. Is it

possible to use this() and super() both in same constructor? 31.

What is the meaning of object cloning in Java? Static

32. In Java, why do we use static variable?

33. Why it is not a good practice to create static variables in

Java? 34. What is the purpose of static method in Java?

35. Why do we mark main method as static in Java?

36. In what scenario do we use a static block?

37. Is it possible to execute a program without defining a main() method?

38. What happens when static modifier is not mentioned in the signature of main method?

39. What is the difference between static method and instance method in Java?

Method Overloading and Overriding

40. What is the other name of Method Overloading?

41. How will you implement method overloading in Java?

42. What kinds of argument variations are allowed in Method Overloading?

43. Why it is not possible to do method overloading by changing return type of method in java?

44. Is it allowed to overload main() method in Java?

45. How do we implement method overriding in Java?

46. Are we allowed to override a static method in Java?

47. Why Java does not allow overriding a static

method? 48. Is it allowed to override an overloaded

method?

49. What is the difference between method overloading and

method overriding in Java?

50. Does Java allow virtual functions?

51. What is meant by covariant return type in Java?

Polymorphism
52. What is Runtime Polymorphism?

53. Is it possible to achieve Runtime Polymorphism by data members in Java?

54. Explain the difference between static and dynamic

binding? Abstraction

55. What is Abstraction in Object Oriented

programming? 56. How is Abstraction different from

Encapsulation?

57. What is an abstract class in Java?

58. Is it allowed to mark a method abstract method without marking the class abstract?

59. Is it allowed to mark a method abstract as well as

final? 60. Can we instantiate an abstract class in Java?

61. What is an interface in Java?

62. Is it allowed to mark an interface method as static?

63. Why an Interface cannot be marked as final in

Java? 64. What is a marker interface?

65. What can we use instead of Marker interface?

66. How Annotations are better than Marker Interfaces? 67. What is

the difference between abstract class and interface in Java?

68. Does Java allow us to use private and protected modifiers for variables in interfaces?

69. How can we cast to an object reference to an interface reference?

Final

70. How can you change the value of a final variable in

Java? 71. Can a class be marked final in Java?

72. How can we create a final method in Java?

73. How can we prohibit inheritance in Java?

74. Why Integer class in final in Java?

75. What is a blank final variable in Java?

76. How can we initialize a blank final variable?

77. Is it allowed to declare main method as final?

Package

78. What is the purpose of package in Java?

79. What is java.lang package?

80. Which is the most important class in Java?

81. Is it mandatory to import java.lang package every time?

82. Can you import same package or class twice in your

class? 83. What is a static import in Java?

84. What is the difference between import static com.test.Fooclass and import com.test.Fooclass?

Internationalization

85. What is Locale in Java?

86. How will you use a specific Locale in Java?

Serialization

87. What is the serialization?

88. What isthe purpose of serialization?
89. What is Deserialization?

90. What is Serialization and Deserialization

conceptually? 91. Why do we mark a data member

transient?

92. Is it allowed to mark a method as transient?

93. How does marking a field as transient makes it possible to serialize an object?

94. What is Externalizable interface in Java?

95. What is the difference between Serializable and Externalizable interface?

Reflection

96. What is Reflection in Java?

97. What are the uses of Reflection in Java?

98. How can we access private method of a class from outside the class?

99. How can we create an Object dynamically at Runtime in

Java? Garbage Collection

100.What is Garbage Collection in Java?

101. Why Java provides Garbage Collector?

102.What is the purpose of gc() in Java?

103.How does Garbage Collection work in Java?

104.When does an object become eligible for Garbage Collection in Java? 105.Why do we use finalize() method in Java?

106.What are the different types of References in Java?

107.How can we reference an unreferenced object again?

108.What kind of process is the Garbage collector thread? 109.What isthe purpose of the Runtime class?

110.How can we invoke an external process in Java?

111. What are the uses of Runtime class?

Inner Classes

112.What is a Nested class?

113.How many types of Nested classes are in Java?

114.Why do we use Nested Classes?

115.What is the difference between a Nested class and an Inner class in Java?

116.What is a Nested interface?

117.How can we access the non-final local variable, inside a Local Inner class?

118.Can an Interface be defined in a Class?

119.Do we have to explicitly mark a Nested Interface public

static? 120.Why do we use Static Nested interface in Java?

String

121. What is the meaning of Immutable in the context of String class in Java?

122.Why a String object is considered immutable in

java? 123.How many objects does following code

create?

124.How many ways are there in Java to create a String

object? 125.How many objects does following code create?

126.What is String interning?

127.Why Java uses String literal concept?

128.What is the basic difference between a String and StringBuffer object?

129.How will you create an immutable class in Java?

130.What is the use of toString() method in java ?

131. Arrange the three classes String, StringBuffer and StringBuilder in the order of efficiency for String processing operations? Exception Handling

132.What is Exception Handling in Java?

133.In Java, what are the differences between a Checked and Unchecked?

134.What is the base class for Error and Exception classes in

Java? 135.What is a finally block in Java?

136.What is the use of finally block in Java?

137.Can we create a finally block without creating a catch block? 138.Do we have to always put a catch block after a try block? 139.In what scenarios, a finally block will not be executed? 140.Can we re-throw an Exception in Java?

141. What is the difference between throw and throws in Java? 142.What is the concept of Exception Propagation?

143.When we override a method in a Child class, can we throw an additional Exception that is not thrown by the Parent class method? Multi-threading
144.How Multi-threading works in Java?

145.What are the advantages of Multithreading?

146.What are the disadvantages of Multithreading?

147.What is a Thread in Java?

148.What is a Thread's priority and how it is used in scheduling?

149.What are the differences between Pre-emptive Scheduling Scheduler and Time Slicing Scheduler?

150.Is it possible to call run() method instead of start() on a thread in Java?

151. How will you make a user thread into daemon thread if it has already started?

152.Can we start a thread two times in Java?

153.In what scenarios can we interrupt a thread?

154.In Java, is it possible to lock an object for exclusive use by a thread?

155.How notify() method is different from notifyAll()
method? Collections

156.What are the differences between the two data structures: a Vector and an ArrayList?

157.What are the differences between Collection and Collections in

Java? 158.In which scenario, LinkedList is better than ArrayList in

Java? 159.What are the differences between a List and Set collection in

Java?

160.What are the differences between a HashSet and TreeSet collection in Java?

161.In Java, how will you decide when to use a List, Set or a Map collection?

162.What are the differences between a HashMap and a Hashtable in Java?

163.What are the differences between a HashMap and a TreeMap?

164.What are the differences between Comparable and

Comparator? 165.In Java, what is the purpose of Properties file?

166.What is the reason for overriding equals()

method? 167.How does hashCode() method work in

Java?

168.Is it a good idea to use Generics in collections?
Mixed Questions

169.What are Wrapper classes in Java?

170.What is the purpose of native method in Java?

171. What is System class?

172.What is System, out and println in System.out.println method

call? 173.What is the other name of Shallow Copy in Java?

174.What is the difference between Shallow Copy and Deep Copy in Java?

175.What is a Singleton class?

176.What is the difference between Singleton class and Static

class? Java Collection

177.What is the difference between Collection and Collections Framework in Java?

178.What are the main benefits of Collections Framework in

Java? 179.What is the root interface of Collection hierarchy in

Java?

180.What are the main differences between Collection and

Collections? 181. What are the Thread-safe classes in Java Collections

framework?

182.How will you efficiently remove elements while iterating a Collection?

183.How will you convert a List into an array of integers like- int[]?

184.How will you convert an array of primitive integers int[] to a List collection?

185.How will you run a filter on a Collection?

186.How will you convert a List to a Set?

187.How will you remove duplicate elements from an ArrayList?

188.How can you maintain a Collection with elements in Sorted

order?

189.What is the difference between Collections.emptyList() and creating new instance of Collection?

190.How will you copy elements from a Source List to another list? 191.

What are the Java Collection classes that implement List interface?

192.What are the Java Collection classes that implement Set interface?

193.What is the difference between an Iterator and ListIterator in

Java? 194.What is the difference between Iterator and Enumeration?

195.What is the difference between an ArrayList and a LinkedList data structure?

196.What is the difference between a Set and a Map in

Java? 197.What is the use of a Dictionary class?

198.What is the default size of load factor in a HashMap collection in Java?
199.What is the significance of load factor in a HashMap in Java?

200.What are the major differences between a HashSet and a HashMap?

201.What are the similarities between a HashSet and a HashMap in

Java? 202.What is the reason for overriding equals() method?

203. How can we synchronize the elements of a List, a Set or a Map?

204.What is Hash Collision? How Java handles hash-collision in HashMap?

205.What are the Hash Collision resolution techniques?

206.What is the difference between Queue and Stack data

structures? 207.What is an Iterator in Java?

208.What is the difference between Iterator and Enumeration in Java?

209.What is the design pattern used in the implementation of Enumeration in Java?

210. Which methods do we need to override to use an object as key in a HashMap?

211. How will you reverse a List in Java?

212. How will you convert an array of String objects into a List?

213. What is the difference between peek(), poll() and remove() methods of Queue interface in java?

214. What is the difference between Array and ArrayList in Java?

215. How will you insert, delete and retrieve elements from a HashMap collection in Java?

216. What are the main differences between HashMap and ConcurrentHashMap in Java?
217. What is the increasing order of performance for following collection classes in Java?

218. Why does Map interface not extend Collection interface in

Java? 219. What are the different ways to iterate elements of a list in

Java?

220.What is CopyOnWriteArrayList? How it is different from ArrayList in Java?

221. How remove() method is implemented in a

HashMap? 222.What is BlockingQueue in Java

Collections?

223.How is TreeMap class implemented in Java?

224.What is the difference between Fail-fast and Fail-safe iterator in Java?

225.How does ConcurrentHashMap work in Java?

226.What is the importance of hashCode() and equals() methods?

227.What is the contract of hashCode() and equals() methods in

Java? 228.What is an EnumSet in Java?

229.What are the main Concurrent Collection classes in Java? 230. How

will you convert a Collection to SynchronizedCollection in Java? 231.

How IdentityHashMap is different from a regular Map in Java?

232.What is the main use of IdentityHashMap?

233.How can we improve the performance of

IdentityHashMap? 234.Is IdentityHashMap thread-safe?

235.What is a WeakHashMap in Java?

236. How can you make a Collection class read Only in Java?
237.When is UnsupportedOperationException thrown in Java?

238.Let say there is a Customer class. We add objects of Customer class to an ArrayList. How can we sort the Customer objects in

ArrayList by using customer firstName attribute of Customer class?

239.What is the difference between Synchronized Collection and Concurrent Collection?

240.What is the scenario to use ConcurrentHashMap in

Java? 241. How will you create an empty Map in Java?

242.What is the difference between remove() method of Collection and remove() method of Iterator?

243.Between an Array and ArrayList, which one is the preferred collection for storing objects?

244.Is it possible to replace Hashtable with ConcurrentHashMap in Java?

245.How CopyOnWriteArrayList class is different from ArrayList and Vector classes?

246.Why ListIterator has add() method but Iterator does not have?

247.Why do we sometime get ConcurrentModificationException during iteration?

248.How will you convert a Map to a List in Java?

249.How can we create a Map with reverse view and lookup in

Java? 250.How will you create a shallow copy of a Map?

251.Why we cannot create a generic array in Java?

252.What is a PriorityQueue in Java?

253.What are the important points to remember while using Java Collections Framework?

254.How can we pass a Collection as an argument to a method and ensure that method will not be able to modify it?

255.Can you explain how HashMap works in Java?

256.Can you explain how HashSet is implemented in

Java? 257.What is a NavigableMap in Java?

258.What is the difference between descendingKeySet() and descendingMap() methods of NavigableMap?

259.What is the advantage of NavigableMap over Map?

260.What is the difference between headMap(), tailMap() and subMap() methods of NavigableMap?

261. How will you sort objects by Natural order in a Java

List? 262.How can we get a Stream from a List in Java?

263. Can we get a Map from a Stream in Java?

264.What are the popular implementations of Deque in

Java? Advanced Multi-threading

265.What is a Thread in Java?

266.What is the priority of a Thread and how it is used in

scheduling? 267.What is the default priority of a thread in Java?

268.What are the three different priorities that can be set on a Thread in Java?

269.What is the purpose of join() method in Thread class?

270.What is the fundamental difference between wait() and sleep() methods?

271. Is it possible to call run() method instead of start() on a thread in Java?

272.What is a daemon thread in Java?

273. How can we make a regular thread Daemon thread in Java?

274.How will you make a user thread into daemon thread if it has already started?

275.Can we start a thread two times in Java?

276.What is a Shutdown hook in Java?

277.What is synchronization in Java?

278.What is the purpose of Synchronized block in

Java? 279.What is static synchronization?

280.What is a Deadlock situation?

281. What is the meaning of concurrency?

282.What is the main difference between process and

thread? 283.What is a process and thread in the context of

Java? 284.What is a Scheduler?

285.What is the minimum number of Threads in a Java

program? 286.What are the properties of a Java thread?

287.What are the different states of a Thread in Java?

288.How will you set the priority of a thread in Java?

289.What is the purpose of Thread Groups in Java?

290.Why we should not stop a thread by calling its stop()

method? 291. How will you create a Thread in Java?

292.How can we stop a thread in the middle of execution in

Java? 293.How do you access the current thread in a Java

program? 294.What is Busy waiting in Multi-threading?

295.How can we prevent busy waiting in Java?

296.Can we use Thread.sleep() method for real-time processing in Java?

297.Can we wake up a thread that has been put to sleep by using Thread.sleep() method?

298.What are the two ways to check if a Thread has been interrupted?

299.How can we make sure that Parent thread waits for termination of Child thread?

300. How will you handle InterruptedException in Java? 301. Which

intrinsic lock is acquired by a synchronized method in Java? 302.Can

we mark a constructor as synchronized in Java? 303.Can we use

primitive values for intrinsic locks?

304.Do we have re-entrant property in intrinsic locks?

305.What is an atomic operation?

306.Can we consider the statement i++ as an atomic operation in

Java? 307.What are the Atomic operations in Java?

308.Can you check if following code is thread-safe?

309.What are the minimum requirements for a Deadlock situation in a program?

310. How can we prevent a Deadlock?

311. How can we detect a Deadlock situation?

312. What is a Livelock?

313. What is Thread starvation?

314. How can a synchronized block cause Thread starvation in

Java? 315. What is a Race condition?

316. What is a Fair lock in multi-threading?

317. Which two methods of Object class can be used to implement a Producer Consumer scenario?

318. How JVM determines which thread should wake up on notify()?

319. Check if following code is thread-safe for retrieving an integer value from a Queue?

320. How can we check if a thread has a monitor lock on a given

object? 321. What is the use of yield() method in Thread class?

322.What is an important point to consider while passing an object from one thread to another thread?

323.What are the rules for creating Immutable

Objects? 324.What is the use of ThreadLocal class?

325.What are the scenarios suitable for using ThreadLocal class?

326. How will you improve the performance of an application by multi threading?

327.What is scalability in a Software program?

328.How will you calculate the maximum speed up of an application by using multiple processors?

329.What is Lock contention in multi-threading?

330.What are the techniques to reduce Lock contention?
331.What technique can be used in following code to reduce Lock contention?

332.What is Lock splitting technique?

333.Which technique is used in ReadWriteLock class for reducing Lock contention?

334.What is Lock striping?

335.What is a CAS operation?

336.Which Java classes use CAS operation?

337.Is it always possible to improve performance by object pooling in a multi-threading application?

338.How can techniques used for performance improvement in a single thread application may degrade the performance in a multi-threading application?

339.What is the relation between Executor and ExecutorService interface?

340.What will happen on calling submit() method of an ExecutorService instance whose queue is already full?

341. What is a ScheduledExecutorService?

342.How will you create a Thread pool in Java?

343.What is the main difference between Runnable and Callable interface?

344.What are the uses of Future interface in Java?

345.What is the difference in concurrency in HashMap and in

Hashtable? 346.How will you create synchronized instance of List or Map Collection? 347.What is a Semaphore in Java?

348.What is a CountDownLatch in Java?

349.What is the difference between CountDownLatch and CyclicBarrier? 350.What are the scenarios suitable for using Fork/Join framework?

351.What is the difference between RecursiveTask and RecursiveAction class?

352.In Java 8, can we process stream operations with a Thread pool? 353.What are the scenarios to use parallel stream in Java 8?

354.How Stack and Heap work in Java multi-threading environment? 355.How can we take Thread dump in Java?

356.Which parameter can be used to control stack size of a thread in Java?

357.There are two threads T1 and T2? How will you ensure that these threads run in sequence T1, T2 in Java?

Java 8

358.What are the new features released in Java 8?

359.What are the main benefits of new features introduced in Java 8? 360.What is a Lambda expression in Java 8?

361.What are the three main parts of a Lambda expression in Java? 362.What is the data type of a Lambda expression?

363.What is the meaning of following lambda expression?

364.Why did Oracle release a new version of Java like Java

8? 365.What are the advantages of a lambda expression?

366.What is a Functional interface in Java 8?
367.What is a Single Abstract Method (SAM) interface in Java

8? 368.How can we define a Functional interface in Java 8?

369.Why do we need Functional interface in Java?

370.Is it mandatory to use @FunctionalInterface annotation to define a Functional interface in Java 8?

371. What are the differences between Collection and Stream API in Java 8?

372.What are the main uses of Stream API in Java 8?

373.What are the differences between Intermediate and Terminal Operations in Java 8 Streams?

374.What is a Spliterator in Java 8?

375.What are the differences between Iterator and Spliterator in Java

8? 376.What is Type Inference in Java 8?

377.Does Java 7 support Type Inference?

378.How does Internal Iteration work in Java 8?

379.What are the main differences between Internal and External Iterator?

380.What are the main advantages of Internal Iterator over External Iterator in Java 8?

381. What are the applications in which we should use Internal Iteration?

382.What is the main disadvantage of Internal Iteration over External Iteration?

383.Can we provide implementation of a method in a Java

Interface? 384.What is a Default Method in an Interface?

385.Why do we need Default method in a Java 8 Interface?

386.What is the purpose of a Static method in an Interface in Java

8? 387.What are the core ideas behind the Date/Time API of Java

8?

388.What are the advantages of new Date and Time API in Java 8 over old Date API?

389.What are the main differences between legacy Date/Time API in Java and Date/Time API of Java 8?

390. How can we get duration between two dates or time in Java 8?

391. What is the new method family introduced in Java 8 for processing of Arrays on multi core machines?

392.How does Java 8 solve Diamond problem of Multiple Inheritance?

393.What are the differences between Predicate, Supplier and Consumer in Java 8?

394.Is it possible to have default method definition in an interface without marking it with default keyword?

395. Can we create a class that implements two Interfaces with default methods of same name and signature?

396. How Java 8 supports Multiple Inheritance?

397.In case we create a class that extends a base class and implements

an interface. If both base class and interface have a default method with same name and arguments, then which definition will be picked by JVM?

398.If we create same method and define it in a class , in its parent class and in an interface implemented by the class, then definition will be invoked if we access it using the reference of Interface and the object of class?

399.Can we access a static method of an interface by using reference of the interface?

400.How can you get the name of Parameter in Java by using

reflection? 401. What is Optional in Java 8?

402.What are the uses of Optional?

403.Which method in Optional provides the fallback mechanism in case of null value?

404.How can we get current time by using Date/Time API of Java 8?

405.Is it possible to define a static method in an Interface? 406.How can

we analyze the dependencies in Java classes and packages? 407.What

are the new JVM arguments introduced by Java 8? 408.What are the

popular annotations introduced in Java 8? 409.What is a StringJoiner

in Java 8?

410.What is the type of a Lambda expression in Java

8? 411.What is the target type of a lambda expression

?

412.What are the main differences between an interface with default method and an abstract class in Java 8?

Java Tricky Questions

413.Is there any difference between a = a + b and a += b expressions?

414.What does the expression 1.0 / 0.0 return? Will there be any compilation error?

415.Can we use multiple main methods in multiple classes?

416.Does Java allow you to override a private or static

method?

417.What happens when you put a key object in a HashMap that is already present?

418.How can you make sure that N threads can access N resources without deadlock?

419.How can you determine if JVM is 32-bit or 64-bit from Java Program?

420.What is the right data type to represent Money (like Dollar/Pound) in Java?

421.How can you do multiple inheritances in Java?

422.Is ++ operation thread-safe in Java?

423.How can you access a non-static variable from the static context?

424.Let say there is a method that throws NullPointerException in the superclass. Can we override it with a method that throws RuntimeException?

425.How can you mark an array volatile in Java?

426.What is a thread local variable in Java?

427.What is the difference between sleep() and wait() methods in Java?

428.Can you create an Immutable object that contains a mutable

object? 429.How can you convert an Array of bytes to String?

430.What is difference between CyclicBarrier and CountDownLatch class?

431.What is the difference between StringBuffer and

StringBuilder? 432.Which class contains clone method? Cloneable

or Object class? 433.How will you take thread dump in Java?

434.Can you cast an int variable into a byte variable? What happens if the value of int is larger than byte?

435.In Java, can we store a double value in a long variable without explicit casting?

436.What will this return 5*0.1 == 0.5? true or false?

437.Out of an int and Integer, which one takes more

memory? 438.Can we use String in the switch case

statement in Java? 439.Can we use multiple main methods

in same class?

440.When creating an abstract class, is it a good idea to call abstract methods inside its constructor?

441. How can you do constructor chaining in Java?

442.How can we find the memory usage of JVM from Java code?

443.What is the difference between x == y and x.equals(y) expressions in Java?

444. How can you guarantee that the garbage collection takes place?

445.What is the relation between x.hashCode() method and x.equals(y) method of Object class?

446.What is a compile time constant in Java?

447.Explain the difference between fail-fast and fail-safe iterators?

448. You have a character array and a String. Which one is more secure to store sensitive data (like password, date of birth, etc.)? 449.Why do you use volatile keyword in Java?

450.What is the difference between poll() and remove() methods of Queue in Java?

451. Can you catch an exception thrown by another thread in Java?

452.How do you decide which type of Inner Class – Static or Non-Static to use in Java?

453.What are the different types of Classloaders in Java? 454.What are the situations in which you choose HashSet or

TreeSet? 455.What is the use of method references in Java?

456.Do you think Java Enums are more powerful than integer

constants? 457.Why do we use static initializers in Java?

458.Your client is complaining that your code is throwing NoClassDefFoundError or NoSuchMethodError, even though you are able to compile your code without error and method exists in your code. What could be the reason behind this?

459.How can you check if a String is a number by using regular expression?

460.What is the difference between the expressions String s = "Temporary" and String s = new String("Temporary ")? Which one is better and more efficient?

461.In Java, can two equal objects have the different hash

code? 462.How can we print an Array in Java?

463.Is it ok to use random numbers in the implementation of hashcode() method in Java?

464.Between two types of dependency injections, constructor injection and setter dependency injection, which one is better?

465.What is the difference between DOM and SAX parser in Java?

466.Between Enumeration and Iterator, which one has better performance in Java?

467.What is the difference between pass by reference and pass by

value? 468.What are the different ways to sort a collection in Java?

469.Why Collection interface doesn’t extend Cloneable and Serializable interfaces?
470.What is the difference between a process and a thread in Java?

471. What are the benefits of using an unordered array over an ordered array?

472.Between HashSet and TreeSet collections in Java, which one is better?

473.When does JVM call the finalize() method?

474.When would you use Serial Garabage collector or Throughput Garbage collector in Java?

475.In Java, if you set an object reference to null, will the Garbage Collector immediately free the memory held by that object?

476.How can you make an Object eligible for Garbage collection in Java?

477.When do you use Exception or Error in Java? What is the difference between these two?

478.What is the advantage of PreparedStatement over Statement class in Java?

479.In Java, what is the difference between throw and throws keywords?

480.What happens to the Exception object after the exception handling is done?

481. How do you find which client machine is sending request to your servlet in Java?

482.What is the difference between a Cookie and a Session object in Java?

483.Which protocol does Browser and Servlet use to communicate with each other?

484. What is HTTPTunneling?

485.Why do we use JSP instead of Servlet in Java?
486.Is empty '.java'file name a valid source file name in

Java? 487.How do you implement Servlet Chaining in Java?

488.Can you instantiate this class?

489.Why Java does not support operator overloading?

490.Why String class is Immutable or Final in Java?

491.What is the difference between sendRedirect and forward methods?

492.How do you fix your Serializable class, if it contains a member that is not serializable?

493.What is the use of run time polymorphism in Java?

494.What are the rules of method overloading and method overriding in Java?

495.What is the difference between a class and an object in Java?

496.Can we create an abstract class that extends another abstract

class? 497.Why do you use Upcasting or Downcasting in Java ?

498.What is the reason to organize classes and interfaces in a package in Java?

499.What is information hiding in Java?

500.Why does Java provide default constructor?

501.What is the difference between super and this keywords in

Java? 502.What is the advantage of using Unicode characters in

Java? 503.Can you override an overloaded method in Java?

504.How can we change the heap size of a JVM?

505.Why should you define a default constructor in Java?
506.How will you make an Object Immutable in Java?

507.How can you prevent SQL Injection in Java Code?

508.Which two methods should be always implemented by HashMap key Object?

509.Why an Object used as Key in HashMap should be

Immutable? 510. How can we share an object between multiple

threads? 511. How can you determine if your program has a

deadlock? JSP

512. What are the implicit objects in JSP?

513. How will you extend JSP code?

514. How will you handle runtime exceptions in JSP?

515. How will you prevent multiple submits of a page that come by clicking refresh button multiple times?

516. How will you implement a thread safe JSP page?

517. How will you include a static file in a JSP page?

518. What are the lifecycle methods of a JSP?

519. What are the advantages of using JSP in web

architecture? 520.What is the advantage of JSP over

Javascript?

521.What is the Lifecycle of JSP?

522.What is a JSP expression?

523.What are the different types of directive tags in

JSP? 524.What is session attribute in JSP?

525.What are the different scopes of a JSP object?
526.What is pageContext in JSP?

527.What is the use of jsp:useBean in JSP?

528.What is difference between include Directive and include Action of JSP?

529.How will you use other Java files of your application in JSP

code? 530.How will you use an existing class and extend it to use in

the JSP? 531.Why _jspService method starts with _ symbol in JSP?

532.Why do we use tag library in JSP?

533.What is the different type of tag library groups in JSTL?

534.How will you pass information from one JSP to another

JSP? 535.How will you call a stored procedure from JSP?

536.Can we override _jspService() method in JSP?

537.What is a directive in JSP?

538.How will you implement Session tracking in JSP?

539. How do you debug code in JSP?

540.How will you implement error page in JSP?

541. How will you send XML data from a JSP?

542.What happens when we request for a JSP page from web

browser? 543.How will you implement Auto Refresh of page in JSP?

544.What are the important status codes in HTTP?

545.What is the meaning of Accept attribute in HTTP header?

546.What is the difference between Expression and Scriptlet in

JSP?

547.How will you delete a Cookie in JSP?

548.How will you use a Cookie in JSP?

549.What is the main difference between a Session and Cookie in

JSP? 550.How will you prevent creation of session in JSP?

551.What is an output comment in JSP?

552.How will you prevent caching of HTMLoutput by web browser in JSP?

553.How will you redirect request to another page in browser in JSP code?

554.What is the difference between sendRedirect and forward in a

JSP? 555.What is the use of config implicit object in JSP?

556.What is the difference between init-param and

context-param? 557.What is the purpose of RequestDispatcher?

558.How can be read data from a Form in a JSP?

559.What is a filter in JSP?

560.How can you upload a large file in JSP?

561.In which scenario, Container initializes multiple JSP/Servlet

objects? Java Design Patterns

562.When will you use Strategy Design Pattern in

Java? 563.What is Observer design pattern?

564.What are the examples of Observer design pattern in JDK?

565.How Strategy design pattern is different from State design pattern in Java?
566.Can you explain Decorator design pattern with an example in

Java? 567.What is a good scenario for using Composite design Pattern

in Java? 568.Have you used Singleton design pattern in your Java

project? 569.What are the main uses of Singleton design pattern in

Java project? 570.Why java.lang.Runtime is a Singleton in Java?

571.What is the way to implement a thread-safe Singleton design pattern in Java?

572.What are the examples of Singleton design pattern in

JDK? 573.What is Template Method design pattern in Java?

574.What are the examples of Template method design pattern in JDK?

575.Can you tell some examples of Factory Method design pattern implementation in Java?

576.What is the benefit we get by using static factory method to create object?

577.What are the examples of Builder design pattern in JDK?

578.What are the examples of Abstract Factory design pattern in

JDK? 579.What are the examples of Decorator design pattern in

JDK? 580.What are the examples of Proxy design pattern in JDK?

581. What are the examples of Chain of Responsibility design pattern in JDK?

582.What are the main uses of Command design pattern?

583.What are the examples of Command design pattern in

JDK? 584.What are the examples of Interpreter design pattern

in JDK?

585.What are the examples of Mediator design pattern in JDK?

586.What are the examples of Strategy design pattern in JDK?

587.What are the examples of Visitor design pattern in JDK?

588.How Decorator design pattern is different from Proxy

pattern?

589.What are the different scenarios to use Setter and Constructor based injection in Dependency Injection (DI) design pattern?

590.What are the different scenarios for using Proxy design pattern?

591. What is the main difference between Adapter and Proxy design pattern?

592.When will you use Adapter design pattern in Java?

593.What are the examples of Adapter design pattern in

JDK?

594.What is the difference between Factory and Abstract Factory design pattern?

595.What is Open/closed design principle in Software

engineering? 596.What is SOLID design principle?

597.What is Builder design pattern?

598.What are the different categories of Design Patterns used in Object Oriented Design?

599.What is the design pattern suitable to access elements of a Collection?

600. How can we implement Producer Consumer design pattern in Java?

601. What design pattern is suitable to add new features to an existing object?

602.Which design pattern can be used when to decouple abstraction from the implementation?

603.Which is the design pattern used in Android applications?

604.How can we prevent users from creating more than one instance of singleton object by using clone() method?

605.What is the use of Interceptor design pattern?

606.What are the Architectural patterns that you have

used? 607.What are the popular uses of Façade design

pattern?

608.What is the difference between Builder design pattern and Factory design pattern?

609.What is Memento design pattern?

610.What is an AntiPattern?

611. What is a Data Access Object (DAO) design
pattern? Spring Questions

612.What is Spring framework?

613.What are the benefits of Spring framework in software

development? 614.What are the modules in Core Container of Spring

framework?

615. What are the modules in Data Access/Integration layer of Spring framework?

616.What are the modules in Web layer of Spring framework?

617.What is the main use of Core Container module in Spring

framework? 618.What kind of testing can be done in Spring Test

Module? 619.What is the use of BeanFactory in Spring framework?

620.Which is the most popular implementation of BeanFactory in

Spring? 621.What is XMLBeanFactory in Spring framework?

622.What are the uses of AOP module in Spring framework?

623.What are the benefits of JDBC abstraction layer module in Spring framework?

624.How does Spring support Object Relational Mapping (ORM) integration?

625.How does Web module work in Spring framework?

626.What are the main uses of Spring MVC module?

627.What is the purpose of Spring configuration file?

628.What is the purpose of Spring IoC container?

629.What is the main benefit of Inversion of Control (IOC) principle?

630. Does IOC containers support Eager Instantiation or Lazy loading of beans?

631. What are the benefits of ApplicationContext in Spring? 632.

How will you implement ApplicationContext in Spring framework?

633.Explain the difference between ApplicationContext and BeanFactory in Spring?

634.Between ApplicationContext and BeanFactory which one is preferable to use in Spring?

635.What are the main components of a typical Spring based

application? 636.Explain Dependency Injection (DI) concept in Spring

framework? 637.What are the different roles in Dependency Injection

(DI)?

**638.Spring framework provides what kinds of Dependency
Injection mechanism?**

**639.In Spring framework, which Dependency Injection is
better? Constructor-based DI or Setter-based DI?
640.What are the advantages of Dependency Injection (DI)?**

641.What are the disadvantages of Dependency Injection

(DI)? 642.What is a Spring Bean?

643.What does the definition of a Spring Bean contain?

**644.What are the different ways to provide configuration metadata to
a Spring Container?**

645.What are the different scopes of a Bean supported by

Spring? 646.How will you define the scope of a bean in Spring?

**647.Is it safe to assume that a Singleton bean is thread safe in
Spring Framework?**

648.What are the design-patterns used in Spring framework?

649.What is the lifecycle of a Bean in Spring framework?

650.What are the two main groups of methods in a Bean's

lifecycle? 651.Can we override main lifecycle methods of a Bean

in Spring? 652.What are Inner beans in Spring?

653.How can we inject a Java Collection in Spring

framework? 654.What is Bean wiring in Spring?

655.What is Autowiring in Spring?

656.What are the different modes of Autowiring supported by Spring?

657.What are the cases in which Autowiring may not work in Spring framework?

658.Is it allowed to inject null or empty String values in

Spring? 659.What is a Java-based Configuration in Spring?

660.What is the purpose of @Configuration annotation?

661.What is the difference between Full @Configuration and 'lite' @Beans mode?

662.In Spring framework, what is Annotation-based container configuration?

663.How will you switch on Annotation based wiring in

Spring? 664.What is @Autowired annotation?

665.What is @Required annotation?

666.What are the two ways to enable RequiredAnnotationBeanPostProcessor in Spring?

667.What is @Qualifier annotation in Spring?

668.How Spring framework makes JDBC coding easier for

developers? 669.What is the purpose of JdbcTemplate?

670.What are the benefits of using Spring DAO?

671. What are the different ways to use Hibernate in Spring?

672.What types of Object Relational Mapping (ORM) are supported by Spring?

673. How will you integrate Spring and Hibernate by using HibernateDaoSupport?

674.What are the different types of the Transaction Management supported by Spring framework?

675.What are the benefits provided by Spring Framework's Transaction Management?

676. Given a choice between declarative and programmatic Transaction Management, which method will you choose?
677.What is Aspect Oriented Programming (AOP)

678.What is an Aspect in Spring?

679.In Spring AOP, what is the main difference between a Concern and a Cross cutting concern?

680.What is a Joinpoint in Spring AOP?

681.What is an Advice in Spring AOP?

682.What are the different types of Advice in Spring

AOP? 683.What is a Pointcut in Spring AOP?

684.What is an Introduction in Spring AOP?

685.What is a Target object in Spring AOP?

686.What is a Proxy in Spring AOP?

687.What are the different types of AutoProxy creators in

Spring? 688.What is Weaving in Spring AOP?

689.In Spring AOP, Weaving is done at compile time or run time?

690.What is XML Schema-based Aspect implementation? 691.What

is Annotation-based aspect implementation in Spring AOP?

692.How does Spring MVC framework work?

693.What is DispatcherServlet?

694.Can we have more than one DispatcherServlet in Spring

MVC? 695.What is WebApplicationContext in Spring MVC?

696.What is Controller in Spring MVC framework?

697.What is @RequestMapping annotation in Spring?
698.What are the main features of Spring MVC?

699.What is the difference between a Singleton and Prototype bean in Spring?

700. How will you decide which scope- Prototype or Singleton to use for a bean in Spring?

701. What is the difference between Setter and Constructor based Dependency Injection (DI) in Spring framework?

702.What are the drawbacks of Setter based Dependency Injection (DI) in Spring?

703.What are the differences between Dependency Injection (DI) and Factory Pattern?

704.In Spring framework, what is the difference between FileSystemResource and ClassPathResource?

705. Name some popular Spring framework annotations that you use in your project?

706. How can you upload a file in Spring MVC Application?

707.What are the different types of events provided by Spring framework?

708.What is the difference between DispatcherServlet and ContextLoaderListener in Spring?

709. How will you handle exceptions in Spring MVC

Framework? 710.What are the best practices of Spring

Framework?

711. What is Spring Boot?

Hibernate

712.What is Hibernate framework?
713.What is an Object Relational Mapping (ORM)?

714.What is the purpose of Configuration Interface in

Hibernate? 715.What is Object Relational Impedance

Mismatch?

716.What are the main problems of Object Relational Impedance Mismatch?

717.What are the key characteristics of Hibernate?

718.Can you tell us about the core interfaces of Hibernate framework?

719.How will you map the columns of a DB table to the properties of a Java class in Hibernate?

720.Does Hibernate make it mandatory for a mapping file to have .hbm.xml extension?

721.What are the steps for creating a SessionFactory in

Hibernate? 722.Why do we use POJO in Hibernate?

723.What is Hibernate Query Language (HQL)?

724.How will you call a stored procedure in

Hibernate? 725.What is Criteria API in Hibernate?

726.Why do we use HibernateTemplate?

727.How can you see SQL code generated by Hibernate on console?

728.What are the different types of collections supported by

Hibernate?

729.What is the difference between session.save() and session.saveOrUpdate() methods in Hibernate?

730.What are the advantages of Hibernate framework over

JDBC? 731.How can we get statistics of a SessionFactory in

Hibernate?

732.What is the Transient state of an object in

Hibernate? 733.What is the Detached state of an object

in Hibernate? 734.What is the use of Dirty Checking in

Hibernate?

735.What is the purpose of Callback interface in

Hibernate? 736.What are the different ORM levels in

Hibernate?

737.What are the different ways to configure a Hibernate

application? 738.What is Query Cache in Hibernate?

739.What are the different types of Association mappings supported by Hibernate?

740.What are the different types of Unidirectional Association mappings in Hibernate?

741. What is Unit of Work design pattern?

742.In Hibernate, how can an object go in Detached state? 743.How

will you order the results returned by a Criteria in Hibernate?

744.How does Example criterion work in Hibernate?

745.How does Transaction management work in Hibernate?

746.How can we mark an entity/collection as immutable in

Hibernate?

747.What are the different options to retrieve an object from database in Hibernate?

748.How can we auto-generate primary key in Hibernate? 749.How

will you re-attach an object in Detached state in Hibernate?

750.What is the first level of cache in Hibernate?

751.What are the different second level caches available in Hibernate?
752.Which is the default transaction factory in Hibernate?

753.What are the options to disable second level cache in

Hibernate? 754.What are the different fetching strategies in

Hibernate?

755.What is the difference between Immediate fetching and
Lazy collection fetching?

756.What is ‘Extra lazy fetching’in Hibernate?

757.How can we check is a collection is initialized or not under

Lazy Initialization strategy?

758.What are the different strategies for cache mapping in Hibernate?

759.What is the difference between a Set and a Bag in Hibernate?

760.How can we monitor the performance of Hibernate in an application?

761.How can we check if an Object is in Persistent, Detached or Transient state in Hibernate?

762.What is ‘the inverse side of association’in a mapping? 763.What is ORM metadata?

764.What is the difference between load() and get() method in Hibernate? 765.When should we use get() method or load() method in Hibernate? 766.What is a derived property in Hibernate?

767.How can we use Named Query in Hibernate?

768.What are the two locking strategies in Hibernate?

769.What is the use of version number in Hibernate?

770.What is the use of session.lock() method in Hibernate? 771. What inheritance mapping strategies are supported by Hibernate?

Maven

772.What is Maven?

773.What are the main features of Maven?

774.What areas of a Project can you manage by using Maven? 775.What are the main advantages of Maven?

776.Why do we say “Maven uses convention over configuration”? 777.What are the responsibilities of a Build tool like Maven? 778.What are the differences between Ant and Maven? 779.What is MOJO in Maven?

780.What is a Repository in Maven?

781.What are the different types of repositories in Maven? 782.What is a local repository in Maven?

783.What is a central repository in Maven?

784.What is a Remote repository in Maven?

785.Why we should not store jars in CVS or any other version control system instead of Maven repository?

786.Can anyone upload JARS or artifacts to Central Repository? 787.What is a POM?

788.What is Super POM?

789.What are the main required elements in POM file?

790.What are the phases in Build lifecycle in Maven? 791.What command will you use to package your Maven project?

792.What is the format of fully qualified artifact name of a Maven project?

793.What is an Archetype in Maven?

794.What is the command in Maven to generate an Archetype? 795.What are the three main build lifecycles of

Maven? 796.What are the main uses of a Maven plugin?

797.How will you find the version of a plugin being used?

798.What are the different types of profile in Maven? Where will you define these profiles?

799.What are the different setting files in Maven? Where will you find these files?

800.What are the main elements we can find in settings.xml?

801.How will you check the version of Maven in your

system? 802.How will you verify if Maven is installed on

Windows? 803.What is a Maven artifact?

804.What are the different dependency scopes in

Maven? 805.How can we exclude a dependency in

Maven?

806.How Maven searches for JAR corresponding to a

dependency? 807.What is a transitive dependency in Maven?

808.What are Excluded dependencies in Maven?

809.What are Optional dependencies in Maven?

810.Where will you find the class files after compiling a Maven project successfully?
811.What are the default locations for source, test and build directories in Maven?

812.What is the result of jar:jar goal in Maven?

813.How can we get the debug or error messages from the execution of Maven?

814.What is the difference between a Release version and SNAPSHOT version in Maven?

815.How will you run test classes in Maven?

816.Sometimes Maven compiles the test classes but doesn't run them? What could be the reason for it?

817.How can we skip the running of tests in Maven?

818.Can we create our own directory structure for a project in

Maven? 819.What are the differences between Gradle and Maven?

820.What is the difference between Inheritance and Multi-module in Maven?

821.What is Build portability in Maven?

GIT

822.How can we see n most recent commits in GIT?

823.How can we know if a branch is already merged into master in

GIT? 824.What is the purpose of git stash drop?

825.What is the HEAD in GIT?

826.What is the most popular branching strategy in

GIT? 827.What is SubGit?

828.What is the use of git instaweb?
829.What are git hooks?

830.What is GIT?

831. What is a repository in GIT?

832.What are the main benefits of GIT?

833.What are the disadvantages of GIT?

834.What are the main differences between GIT and

SVN? 835.How will you start GIT for your project?

836.What is git clone in GIT?

837.How will you create a repository in GIT?

838.What are the different ways to start work in GIT?

839.GIT is written in which language?

840.What does 'git pull' command in GIT do

internally? 841.What does 'git push' command in GIT

do internally? 842.What is git stash?

843.What is the meaning of 'stage'in GIT?

844. What is the purpose of git config command?

845.How can we see the configuration settings of GIT

installation? 846.How will you write a message with commit

command in GIT? 847.What is stored inside a commit object in

GIT?

848.How many heads can you create in a GIT

repository? 849.Why do we create branches in GIT?

850.What are the different kinds of branches that can be created in GIT?
851. How will you create a new branch in GIT?

852.How will you add a new feature to the main

branch? 853.What is a pull request in GIT?

854.What is merge conflict in GIT?

855.How can we resolve a merge conflict in GIT?

856.What command will you use to delete a branch?

857.What command will you use to delete a branch that has unmerged changes?

858.What is the alternative command to merging in

GIT? 859.What is Rebasing in GIT?

860.What is the 'Golden Rule of Rebasing'in GIT?

861.Why do we use Interactive Rebasing in place of Auto

Rebasing? 862.What is the command for Rebasing in Git?

863.What is the main difference between git clone and git

remote? 864.What is GIT version control?

865.What GUI do you use for working on GIT?

866.What is the use of git diff command in GIT?

867.What is git rerere?

868.What are the three most popular version of git diff

command? 869.What is the use of git status command?

870.What is the main difference between git diff and git

status? 871.What is the use of git rm command in GIT?

872.What is the command to apply a stash?

873.Why do we use git log command?

874.Why do we need git add command in GIT?

875.Why do we use git reset command?

876.What does a commit object contain?

877.How can we convert git log messages to a different format?

878.What are the programming languages in which git hooks can be written?

879.What is a commit message in GIT?

880.How GIT protects the code in a repository?

881.How GIT provides flexibility in version control?

882.How can we change a commit message in GIT?

883.Why is it advisable to create an additional commit instead of amending an existing commit?

884.What is a bare repository in GIT?

885.How do we put a local repository on GitHub

server? 886.How will you delete a branch in GIT?

887.How can we set up a Git repository to run code sanity checks and UAT tests just before a commit?

888.How can we revert a commit that was pushed earlier and is public now?

889.In GIT, how will you compress last n commits into a single

commit? 890.How will you switch from one branch to a new branch

in GIT?

891.How can we clean unwanted files from our working directory in

GIT? 892.What is the purpose of git tag command?

893.What is cherry-pick in GIT?

894.What is shortlog in GIT?

895.How can you find the names of files that were changed in a specific commit?

896.How can we attach an automated script to run on the event of a new commit by push command?

897.What is the difference between pre-receive, update and post-receive hooks in GIT?

898.Do we have to store Scripts for GIT hooks within same repository?

899.How can we determine the commit that is the source of a bug in

GIT? 900.How can we see differences between two commits in GIT?

901.What are the different ways to identify a commit in GIT?

902.When we run git branch <branchname>, how does GIT know the SHA-1 of the last commit?

903.What are the different types of Tags you can create in

GIT? 904.How can we rename a remote repository?

905.Some people use git checkout and some use git co for checkout. How is that possible?

906.How can we see the last commit on each of our branch in

GIT? 907.Is origin a special branch in GIT?

908.How can we configure GIT to not ask for password every time?

909.What are the four major protocols used by GIT for data transfer?

910.What is GIT protocol?

911. How can we work on a project where we do not have push access? 912.What is git grep?

913.How can your reorder commits in GIT?

914.How will you split a commit into multiple commits?

915.What is filter-branch in GIT?

916.What are the three main trees maintained by GIT?

917.What are the three main steps of working GIT?

918. What are ours and theirs merge options in GIT?

919.How can we ignore merge conflicts due to Whitespace? 920.What is git blame?

921.What is a submodule in GIT?

AWS

922.What do you know about AWS Region?

923.What are the important components of IAM?

924.What are the important points about AWS IAM?

925.What are the important features of Amazon S3?

926.What is the scale of durability in Amazon S3?

927.What are the Consistency levels supported by Amazon S3?

928.What are the different tiers in Amazon S3 storage? 929.How will you upload a file greater than 100 megabytes in Amazon S3? 930.What happens to an Object when we delete it from Amazon S3?

931.What is the use of Amazon Glacier?

932.Can we disable versioning on a version-enabled bucket in Amazon S3? 933.What are the use cases of Cross Region Replication Amazon S3?

934.Can we do Cross Region replication in Amazon S3 without enabling versioning on a bucket?

935.What are the different types of actions in Object Lifecycle Management in Amazon S3?

936. How do we get higher performance in our application by using Amazon CloudFront?

937.What is the mechanism behind Regional Edge Cache in Amazon CloudFront?

938.What are the benefits of Streaming content?

939.What is Lambda@Edge in AWS?

940.What are the different types of events triggered by Amazon CloudFront?

941. What is Geo Targeting in Amazon CloudFront?

942.What are the main features of Amazon CloudFront?

943.What are the security mechanisms available in Amazon S3? Cloud Computing

944.What are the benefits of Cloud Computing?

945.What is On-demand computing in Cloud

Computing? 946.What are the different layers of Cloud

computing?

947.What resources are provided by Infrastructure as a Service (IAAS) provider?

948.What is the benefit of Platform as a Service?
949.What are the main advantages of PaaS?

950.What is the main disadvantage of PaaS?

951.What are the different deployment models in Cloud

computing? 952.What is the difference between Scalability and

Elasticity? 953.What is Software as a Service?

954.What are the different types of Datacenters in Cloud computing?

955.Explain the various modes of Software as a Service (SaaS) cloud environment?

956.What are the important things to care about in Security in a cloud environment?

957.Why do we use APIin cloud computing environment?

958.What are the different areas of Security Management in

cloud? 959.What are the main cost factors of cloud based data

center? 960.How can we measure the cloud-based services?

961. How a traditional datacenter is different from a cloud environment?

962.How will you optimize availability of your application in a Cloud environment?

963.What are the requirements for implementing IaaS strategy in Cloud? DOCKER

964.What is Docker?

965.What is the difference between Docker image and Docker container? 966.How will you remove an image from Docker?

967.How is a Docker container different from a hypervisor? 968.Can we write compose file in json file instead of yaml? 969.Can we run multiple apps on one server with Docker? 970.What are the common use cases of Docker?

971.What are the main features of Docker-compose?

972.What is the most popular use of Docker?

973.What is the role of open source development in the popularity of Docker?

UNIX Shell

974.How will you remove all files in current directory? Including the files that are two levels down in a sub-directory.

975.What is the difference between the –v and –x options in Bash shell scripts?

976.What is a Filter in Unix command?

977.What is Kernel in Unix operating system?

978.What is a Shell in Unix OS?

979.What are the different shells in Unix that you know about? 980.What is the first character of the output in ls –l

command ?

981. What is the difference between Multi-tasking and Multi-user environment?

982.What is Command Substitution in Unix?

983.What is an Inode in Unix?

984.What is the difference between absolute path and relative path in Unix file system?

985.What are the main responsibilities of a Unix Shell?
986.What is a Shell variable?

Microservices

987.What is a Microservice?

988.What are the benefits of Microservices architecture?

989.What is the role of architect in Microservices

architecture?

990.What is the advantage of Microservices architecture over Service Oriented Architecture (SOA)?

991.Is it a good idea to provide a Tailored Service Template for Microservices development in an organization?

992.What are the disadvantages of using Shared libraries approach to decompose a monolith application?

993.What are the characteristics of a Good

Microservice? 994.What is Bounded Context?

995.What are the points to remember during integration of Microservices?

996.Is it a good idea for Microservices to share a common database?

997.What is the preferred type of communication between Microservices? Synchronous or Asynchronous?

998.What is the difference between Orchestration and Choreography in Microservices architecture?

999.What are the issues in using REST over HTTP for Microservices? 1000. Can we create Microservices as State Machines?

ACKNOWLEDGMENTS

We thank our readers who constantly send feedback and reviews to motivate usin creating these useful books with the latest information!

INTRODUCTION

Java is one of the most popular programming language. There is a growing demand for Java Developer jobs in technology companies.

This book contains technical interview questions that an interviewer asks for Java technology and related topics like Spring, Hibernate, Maven, Git, Microservices, AWS etc.

Each question is accompanied with an answer so that you can prepare for job interview in short time.

We have compiled this list after attending dozens of technical interviews in top-notch companies like- Facebook, Oracle, Netflix, Amazon etc.

Once you go through themin the first pass, mark the questions that you could not answer by yourself. Then, in second pass go through only the difficult questions.

After going through this book 2-3 times, you will be well prepared to face a technical interview for a Java Developer position from Software Engineer level to Principal Engineer level.

All the best!!

Java Interview Questions

Java Basics

1. What is the difference between JDK and JRE?

JDK stands for Java Development Kit. It contains the tools and libraries for development of Java programs. It also contains compilers and debuggers needed to compile Java program,

JRE stands for Java Runtime Environment. This is included in JDK. JRE provides libraries and JVM that is required to run a Java program.

2. What is Java Virtual Machine (JVM)?

Java Virtual Machine (JVM) is an abstract machine that executes Java Bytecode. There are different JVM for different hardware and software platforms. So JVM is platform dependent. JVM is responsible for loading, verifying and executing the Bytecode on a platform.

3. What are the different types of memory areas allocated by JVM?

In java, JVM allocates memory to different processes, methods and objects. Some of the memory areas allocated by JVM are:

1. ClassLoader: It is a component of JVM used to load class files.
2. Class (Method) Area: It stores per-class structures such as the runtime constant pool, field and method data, and the code for methods.
3. Heap: Heap is created a runtime and it contains the runtime data area in which objects are allocated.
4. Stack: Stack stores local variables and partial results at runtime. It also helps in method invocation and return value. Each thread creates a private JVM stack at the time of thread creation.
5. Program Counter Register: This memory area contains the address of the Java virtual machine instruction that is currently being executed.
6. Native Method Stack: This area is reserved for all the native methods used in the application.

4. What is JIT compiler?

Just In Time compiler also known as JIT compiler is used for performance improvement in Java. It is enabled by default. It is compilation done at execution time rather earlier.
Java has popularized the use of JIT compiler by including it in JVM.

5. How Java platform is different from other platforms?

Java is a platform independent language. Java compiler converts Java code in to byte code that can be interpreted by JVM. There are JVM written for almost all the popular platforms in the world.

Java byte code can run on any supported platform in same way. Where as other languages require libraries compiled for a specific platform to run.

6. Why people say that Java is 'write once and run anywhere' language?

You can write Java code on Windows and compile it in Windows platform. The class and jar files that you get from Windows platform can run as it is on Unix environment. So it is a truly platformindependent language.

Behind all this portability is Java byte code. Byte code generated by Java compiler can be interpreted by any JVM. So it becomes much easier to write programs in Java and expect those to run on any platform.

Java compiler javac compiles java code and JVM java runs that code.

7. How does ClassLoader work in Java?

In Java, ClassLoader is a class that is used to load files in JVM. ClassLoader loads files from their physical file locations e.g.

Filesystem, Network location etc.

There are three main types of ClassLoaders in Java.

1. Bootstrap ClassLoader: This is the first ClassLoader. It loads classes from rt.jar file.
2. Extension ClassLoader: It loads class files from jre/lib/ext location.
3. Application ClassLoader: This ClassLoader depends on CLASSPATH to find the location of class files. If you specify your jars in CLASSPATH, then this ClassLoader will load them.

8. Do you think 'main' used for main method is a keyword in Java?

No, main is just a name of method. There can be multiple methods with same name main in a class file. It is not a keyword in Java.

9. Can we write main method as public void static instead of public static void?

No, you cannot write it like this. Any method has to first specify the modifiers and then the return value. The order of modifiers can change.

We can write static public void main() instead of public static void main().

10.In Java, if we do not specify any value for local variables, then what

will be the default value of the local variables?

Java does not initialize local variables with any default value. So these variables will be just null by default.

11.Let say, we run a java class without passing any arguments. What will be the value of String array of arguments in Main method?

By default, the value of String array of arguments is empty in Java. It is not null.

12.What is the difference between byte and char data types in Java?

Both byte and char are numeric data types in Java. They are used to represent numbers in a specific range.

Major difference between them is that a byte can store raw binary data where as a char stores characters or text data.

Usage of char is E.g. char ch = ‘x’;

Byte values range from -128 to 127.

A byte is made of 8 bits. But a char is made of 16 bits. So it is equivalent to 2 bytes.

OOPS

13.What are the main principles of Object Oriented Programming?

Main principles of Object Oriented Programming (OOPS) are: 1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

14.What is the difference between Object Oriented Programming language and Object Based Programming language?

Object Oriented Programming languages like Java and C++ follow concepts of OOPS like- Encapsulation, Abstraction, Polymorphism and Inheritance etc.

Object Based Programming languages follow some features of OOPS but they do not provide support for Polymorphism and Inheritance. Egg. JavaScript, VBScript etc.

Object Based Programming languages provide support for Objects and you can build objects from constructor. They languages also support Encapsulation. These are also known as Prototype-oriented languages.

15.In Java what is the default value of an object reference defined as an

instance variable in an Object?

All the instance variable object references in Java are null.

16.Why do we need constructor in Java?

Java is an object-oriented language, in which we create and use objects. A constructor is a piece of code similar to a method. It is used to create an object and set the initial state of the object.

A constructor is a special function that has same name as class name.

Without a constructor, there is no other way to create an object.

By default, Java provides a default constructor for every object. If we overload a constructor then we have to implement default constructor.

17.Why do we need default constructor in Java classes?

Default constructor is the no-argument constructor that is automatically generated by Java if no other constructor is defined.

Java specification says that it will provide a default constructor if there is no overloaded constructor in a class. But it does not say anything about the scenario in which we write an overloaded constructor in a class.

We need at least one constructor to create an object, that’s why Java provides a default constructor.

When we have overloaded constructor, then Java assumes that we want some custom treatment in our code. Due to which it does not provide default constructor. But it needs default constructor as per the specification. So it gives error.

18.What is the value returned by Constructor in Java?

When we call a constructor in Java, it returns the object created by it. That is how we create new objects in Java.

19.Can we inherit a Constructor?

No, Java does notsupport inheritance of constructor.

20. Why constructors cannot be final, static, or abstract in Java?

If we set a method as final it means we do not want any class to override it. But the constructor (as per Java Language Specification) cannot be overridden. So there is no use of marking it final.

If we set a method as abstract it means that it has no body and it should be implemented in a child class. But the constructor is called implicitly when the new keyword is used. Therefore it needs a body.

If we set a method as static it means that it belongs to the class, but not a particular object. The constructor is always called to initialize an object. Therefore, there is no use of marking constructor static.

Inheritance

21.What is the purpose of ‘this’ keyword in java?

In Java, ‘this’ keyword refers to current instance of the object.

It is useful for differentiating between instance variables and local variables.

It can be used to call constructors. Or it can be used to refer to the instance.

In case of method overriding, this is used for falling the method of current class.