Core Java FAQs

1. What is the difference between JDK and JRE, JVM and JIT Compiler?

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. What is platform independency in java?

7. How many class loaders in java?

8. What is delegation Hierarchy Algorithm?

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 whatwill be the default value of the local variables?

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

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

13. Ascending order of numeric data types?

14. Can I take multi classes in single java file?

15. OOPS

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

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

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

19. Why do we need constructor in Java?

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

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

22. Can we inherit a Constructor?

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

Inheritance

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

25. Explain the concept of a static variable. How is it different from an instance variable?

26. Explain the concept of Inheritance?

27. Which class in Java is superclass of every other class?

28. Why Java does not support multiple inheritance?

29. In OOPS, what is meant by composition?

30. How aggregation and composition are different concepts?

31. Why there are no pointers in Java?

32. What is encapsulation in Java, and why is it considered a fundamental principle of object-oriented programming?

33. What is the purpose of ‘super’ keyword in java?

34. Is it possible to use this() and super() both in same constructor?

35. What is the meaning of object cloning in Java?

Static

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

37. Why it is not a good practice to create static variables in Java?

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

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

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

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

42. What happens when static modifier is not mentioned in the signatureof main method?

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

Method Overloading and Overriding

44. What is the other name of Method Overloading?

45. How will you implement method overloading in Java?

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

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

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

49. How do we implement method overriding in Java?

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

51. Why Java does not allow overriding a static method?

52. Is it allowed to override an overloaded method?

53. What is the difference between method overloading and method overriding in Java?

54. Does Java allow virtual functions?

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

Polymorphism

56. What is Runtime Polymorphism?

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

58. Explain the difference between static and dynamic binding?

Abstraction

59. What is Abstraction in Object Oriented programming?

60. How is Abstraction different from Encapsulation?

61. What is an abstract class in Java?

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

63. Is it allowed to mark a method abstract as well as final?

64. Can we instantiate an abstract class in Java?

65. What is an interface in Java?

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

67. Why an Interface cannot be marked as final in Java?

68. What is a marker interface?

69. What can we use instead of Marker interface?

70. How Annotations are better than Marker Interfaces?

71. What is the difference between abstract class and interface in Java?

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

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

Final

74. How can you change the value of a final variable in Java?

75. Can a class be marked final in Java?

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

77. How can we prohibit inheritance in Java?

78. Why Integer class in final in Java?

79. What is a blank final variable in Java?

80. How can we initialize a blank final variable?

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

Package

82. What is the purpose of package in Java?

83. What is java.lang package?

84. Which is the most important class in Java?

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

86. Can you import same package or class twice in your class?

87. What is a static import in Java?

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

Internationalization

89. What is Locale in Java?

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

Serialization

91. What is the serialization?

92. What is the purpose of serialization?

93. What is Deserialization?

94. What is Serialization and Deserialization conceptually?

95. Why do we mark a data member transient?

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

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

98. What is Externalizable interface in Java?

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

Reflection

100. What is Reflection in Java?

101. What are the uses of Reflection in Java?

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

103. How can we create an Object dynamically at Runtime in Java?

Garbage Collection

104. What is Garbage Collection in Java?

105. Why Java provides Garbage Collector?

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

107. How does Garbage Collection work in Java?

108. When does an object become eligible for Garbage Collection in Java?

109. Why do we use finalize() method in Java?

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

111. How can we reference an unreferenced object again?

112. What kind of process is the Garbage collector thread?

113. What is the purpose of the Runtime class?

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

115. What are the uses of Runtime class?

Inner Classes

116. What is a Nested class?

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

118. Why do we use Nested Classes?

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

120. What is a Nested interface?

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

122. Can an Interface be defined in a Class?

123. Do we have to explicitly mark a Nested Interface public static?

124. Why do we use Static Nested interface in Java?

String

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

126. Why a String object is considered immutable in java?

127. How many objects does following code create?

128. How many ways are there in Java to create a String object?

129. How many objects does following code create?

130. What is String interning?

131. Why Java uses String literal concept?

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

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

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

135. Arrange the three classes String, StringBuffer and StringBuilder inthe order of efficiency for String processing operations?

Exception Handling

136. What is Exception Handling in Java?

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

138. What is the base class for Error and Exception classes in Java?

139. What is a finally block in Java?

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

141. Can we create a finally block without creating a catch block?

142. Do we have to always put a catch block after a try block?

143. In what scenarios, a finally block will not be executed?

144. Can we re-throw an Exception in Java?

145. What is the difference between throw and throws in Java?

146. What is the concept of Exception Propagation?

147. 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

148. How Multi-threading works in Java?

149. What are the advantages of Multithreading?

150. What are the disadvantages of Multithreading?

151. What is a Thread in Java?

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

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

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

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

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

157. In what scenarios can we interrupt a thread?

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

159. How notify() method is different from notifyAll() method?

Collections

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

161. What are the differences between Collection and Collections in Java?

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

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

164. What are the differences between a HashSet and TreeSet collectionin Java?

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

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

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

168. What are the differences between Comparable and Comparator?

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

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

171. How does hashCode() method work in Java?

172. Is it a good idea to use Generics in collections?

Mixed Questions

173. What are Wrapper classes in Java?

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

175. What is System class?

176. What is System, out and println in System.out.println method call?

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

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

179. What is a Singleton class?

180. What is the difference between Singleton class and Static class?

Java Collection

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

182. What are the main benefits of Collections Framework in Java?

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

184. What are the main differences between Collection and Collections?

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

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

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

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

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

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

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

192. How can you maintain a Collection with elements in Sorted order?

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

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

195. What are the Java Collection classes that implement List interface?

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

197. What is the difference between an Iterator and ListIterator in Java?

198. What is the difference between Iterator and Enumeration?

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

200. What is the difference between a Set and a Map in Java?

201. What is the use of a Dictionary class?

202. What is the default size of load factor in a HashMap collection in Java?

203. 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 Enumerationin 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() methodsof 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 ArrayListin 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 classto 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 andVector 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? Can you

explain how HashSet is implemented in Java?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?.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 Java8?

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? What is the purpose of a Static method in an Interface in Java 8?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 overold Date API?

389. What are the main differences between legacy Date/Time API in Javaand 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 Consumerin 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 ofnull 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?

413. What is the Stream API in Java?

414. What are the main advantages of using the Stream API?

415. How do you create a stream?

416. What are intermediate and terminal operations?

417. What is the map() operation?

418. How does filtering work in streams?

419. What is the difference between collect() and reduce()?

420. Can you perform parallel processing with the Stream API?

421. What is the purpose of the flatMap() operation?

422. How can you limit the number of elements in a stream?

423. What does skip() do in a stream?

424. What are primitive streams?

425. What is the difference between findFirst() and findAny()?

426. What is the purpose of the peek() operation?

427. How do you combine multiple streams?

428. What is the allMatch(), anyMatch(), and noneMatch()?

429. What is the Collectors class?

430. How do you use toList()?

431. How can you use toMap() to create a map from a stream?

432. What does the joining() collector do?

433. How do you use groupingBy()?

434. What is partitioningBy()?

435. What is reducing()?