|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q1. Why is String immutable in Java ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. 1. String Pool - When a string is created and if it exists in the pool, the reference of the existing string will be returned instead of creating a new object. If string is not immutable, changing the string with one reference will lead to the wrong value for the other references.  Example -  String str1 = "String1"; String str2 = "String1"; // It doesn't create a new String and rather reuses the string literal from pool  // Now both str1 and str2 pointing to same string object in pool, changing str1 will change it for str2 too  2. To Cache its Hashcode - If string is not immutable, One can change its hashcode and hence it's not fit to be cached.  3. Security - String is widely used as parameter for many java classes, e.g. network connection, opening files, etc. Making it mutable might possess threats due to interception by the other code segment. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **What is the difference between String , StringBuilder and StringBuffer ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | |  | Anonymous | | |
| Ans. If the Object value will not change, use String Class because a String object is immutable.  If the Object value can change and will only be modified from a single thread, use StringBuilder because StringBuilder is unsynchronized(means faster).  If the Object value may change, and can be modified by multiple threads, use a StringBuffer because StringBuffer is thread safe(synchronized). |

**3. What is Lazy Initialization in Hibernate ?**

Ans. It's a feature to lazily initialize dependencies , relationship and associations from the Database. Any related references marked as @OneToMany or @ManyToMany are loaded lazily i.e when they are accessed and not when the parent is loaded.

@Entity

@Table(name = "EMPLOYEE")

**public** **class** **Employee** {

@ManyToOne(fetch = FetchType.LAZY)

**private** Set<Department> dept = **new** Department();

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **4. What is the difference between ArrayList and LinkedList** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. Underlying data structure for ArrayList is Array whereas LinkedList is the linked list and hence have following differences -  1. ArrayList needs continuous memory locations and hence need to be moved to a bigger space if new elements are to be added to a filled array which is not required for LinkedList.  2. Removal and Insertion at specific place in ArrayList requires moving all elements and hence leads to O(n) insertions and removal whereas its constant O(1) for LinkedList.  3. Random access using index in ArrayList is faster than LinkedList which requires traversing the complete list through references.  4. Though Linear Search takes Similar Time for both, Binary Search using LinkedList requires creating new Model called Binary Search Tree which is slower but offers constant time insertion and deletion.  5. For a set of integers you want to sort using quicksort, it's probably faster to use an array; for a set of large structures you want to sort using selection sort, a linked list will be faster. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **What is Polymorphism in Java ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2016-11-06 09:10:54   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. Polymorphism means the condition of occurring in several different forms.  Polymorphism in Java is achieved in two manners   1. Static polymorphism is the polymorphic resolution identified at compile time and is achieved through function overloading whereas   2. Dynamic polymorphism is the polymorphic resolution identified at runtime and is achieved through method overriding. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **6. Can we serialize static variables ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. No. Only Object and its members are serialized. Static variables are shared variables and doesn't correspond to a specific object |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q7. Difference between Vector and ArrayList ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. Vectors are synchronized whereas Array lists are not. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q8. What are various types of Class loaders used by JVM ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | |
| Ans. Bootstrap - Loads JDK internal classes, java.\* packages.  Extensions - Loads jar files from JDK extensions directory - usually lib/ext directory of the JRE  System - Loads classes from system classpath. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q9. What is the use of Transient Keyword ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. It in Java is used to indicate that a field should not be serialized |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **10. What is an immutable class ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. Class using which only immutable (objects that cannot be changed after initialization) objects can be created. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q11. How to implement an immutable class ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | |
| Ans. We can make a class immutable by  1. Making all methods and variables as private.  2. Setting variables within constructor.  Public Class ImmutableClass{   private int member;  ImmutableClass(int var){  member=var;  }  }   and then we can initialize the object of the class as  ImmutableClass immutableObject = new ImmutableClass(5);  Now all members being private , you cant change the state of the object. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q12. Write an algorithm / Java Program to show duplicates in an array of n elements?** | [Algorithm](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Algorithm) | 2017-01-19 15:37:26   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. int duplicateArray[] = { 1, 2, 2, 3, 4, 5, 6, 8, 9} Set unique = new HashSet(); for (int i = 0; i < duplicateArray.length; i) { if (unique.contains(duplicateArray[i])) { System.out.println(duplicateArray[i]); } else { unique.add(duplicateArray[i]); } }  Complexity O(n) = nHashSet contains and add has O(n) = 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **13. What is a binary tree ?** | [Algorithm](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Algorithm) | 2017-01-19 15:38:04   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. Binary tree is a tree in which each node has up to two children.Tree is a data structure composed of nodes.Each tree has a root node(not necessary in graph theory). The root node has zero or more child nodes.Each child node has zero or more child nodes, and so on.The tree cannot contain cycles. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **4. Can we have multiple inheritance in java using abstract class ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2018-07-04 16:17:49   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. No   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **15. Describe structure of a Web application.** | [architecture](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=architecture) | 2018-07-04 16:18:26   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. WEB APP |WEB-INF - META-INF | | | META-INF.MF | lib - WEB.xml - Classes |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **16. Difference between Checked and Unchecked exceptions ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Anonymous | | | | Ans. Checked exceptions are the exceptions for which compiler throws an errors if they are not checked whereas unchecked exceptions are caught during run time only and hence can't be checked. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q17. What is rule regarding overriding equals and hashCode method ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. A Class must override the hashCode method if its overriding the equals method. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q18. Can we have null keys in TreeMap ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. No, results in exception. |   **Q19. Explain Singleton Design Pattern ?**  **1. What is Singleton ?**  Ans. It's a Design Pattern.  **Q2. What is a Singleton Class ?**  Ans. Class using which only one object can be created.  **Q3. What is the use of such a class ?**  Ans. There could be situations where we need not create multiple objects and hence Singleton can help in saving resources by avoiding creating new objects every time a request is made. Moreover these classes can also be helpful if we want a object to be shared among threads.  **Q4. Are u using Singleton in your code ?**  Ans. Yes, for Database connection and Property files.  **Q5. Write the code for a Singleton Class ?**  Ans.  class Singleton {  private static volatile Singleton instance = null;  private Singleton(){}  public static Singleton getInstance() {  if (instance == null) {  synchronized(Singleton.class) {  if (instance== null)  instance = new Singleton();  }  }  return instance;  }  }  **Q6. Why have we used synchronized here ?**  Ans. getInstance method can be accessed from two points simultaneously and in such case 2 instances may get created. Synchronization will make sure that the method gets accessed one by one for each call and the same object is returned for the second call.  **Q7. Why have we declared the instance reference volatile ?**  Ans. That's an instruction to JVM that the variable is getting accessed by multiple locations and hence don't cache it.  **Q8. Can we make the reference instance non static ?**  Ans. No , as non static variables cannot be accessed through static methods.  **Q9. Can we have this pattern implemented using Static Class ?**  Ans. Though we can implement this behavior using static class , but we should never do it.  **Q10. What are the problems in implementing this patterns using static Class ?**  Ans.   a. We cannot achieve runtime Polymorphism or late binding as Java doesn't allow overriding static methods. b. We cannot do lazy initialization as Static members are loaded during class loading only. c. We cannot serialize as Java doesn't serialize static members.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **20. What is ConcurrentModificationException ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. This is the exception that is thrown when we try to modify the non concurrent collection class while iterating through it. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **21. What is ArrayIndexOutOfBoundException ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. Exception thrown by the application is we try to access an element using an index which is not within the range of array i.e lower than 0 or greater than the size of the array. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q22. Difference between jar , war and ear ?** | [Java EE](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Java%20EE) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. Jar is Java Archieve i.e compressed Class or Class / Java files.  War comprises of compressed Servlet class files,JSP FIles,supporting files, GIF and HTML files.  Ear comprise of compressed Java and web module files ( was files ) |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **23. What is Jenkins ?** | [Jenkins](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Jenkins) | |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png | Admin info@buggybread.com | | | | Ans. It is a continuous integration tool written in Java. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **24. What are the disadvantages of multithreading ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2016-10-11 08:47:41   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. 1. Switching Overheads - Even though multi threading aims at improving performance by reducing the wait time and hence improving overall throughput, there is a cost of switching resources between threads and sometime this cost can surpass the benefits if there isnt much wait for IO or external communication.  2. Debugging is hard with multi threaded code.  3. Deadlock - Execution of multi threaded code many a times lead to deadlock due to shared resources. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **25. What is the use of synchronized keyword ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2016-11-04 08:15:53   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. Synchronize is used to achieve mutual exclusion i.e at one time, the segment of the code, method that has been declared synchronized should be executed by single thread only and hence the lock needs to be retrieved before executing the segment and then released. |   [**How is Hashmap in Java implemented internally? What are the pros and cons to use it? What are the complexities it provides for insert, delete and lookup?**](https://www.quora.com/How-is-Hashmap-in-Java-implemented-internally-What-are-the-pros-and-cons-to-use-it-What-are-the-complexities-it-provides-for-insert-delete-and-lookup)  There are four things we should know about before going into internals of how HashMap works -   * **HashMap** works on the principal of hashing. * **Map.Entry interface** - This interface gives a map entry (key-value pair). HashMap in Java stores both key and value object, in bucket, as Entry object which implements this nested interface Map.Entry. * **hashCode()** -HashMap provides put(key, value) for **storing** and get(key) method for**retrieving** Values from HashMap. When put() method is used to store (Key, Value) pair, HashMap implementation **calls hashcode** on Key object to calculate a hash that is used to find a bucket where Entry object will be stored. When get() method is used to retrieve value, again key object is used to calculate a hash which is used then to find a bucket where that particular key is stored. * **equals()** - equals() method is used to **compare objects for equality**. In case of HashMap key object is used for comparison, also using equals() method Map knows how to handle **hashing collision** (hashing collision means more than one key having the same hash value, thus assigned to the same bucket. In that case objects are stored in a linked list. Where hashCode method helps in finding the bucket where that key is stored, equals method helps in finding the right key as there may be more than one key-value pair stored in a single bucket.   ***HashMap changes in Java 8*** Though HashMap implementation provides constant time performance O(1) for get() and put() method but that is in the ideal case when the Hash function distributes the objects evenly among the buckets. But the performance may worsen in the case hashCode() used is not proper and there are lots of hash collisions. As we know now that in case of hash collision entry objects are stored as a node in a linked-list and equals() method is used to compare keys. That comparison to find the correct key with in a linked-list is a linear operation so in a worst case scenario the complexity becomes O(n). To address this issue in Java 8 hash elements use balanced trees instead of linked lists after a certain threshold is reached. Which means HashMap starts with storing Entry objects in linked list but after the number of items in a hash becomes larger than a certain threshold, the hash will change from using a linked list to a balanced tree, this will improve the worst case performance from O(n) to O(log n). For more details refer this - [How HashMap internally works in Java](http://netjs.blogspot.in/2015/05/how-hashmap-internally-works-in-java.html)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **27. What is an exception and exception handling in Java ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2017-01-09 13:01:40   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. An Exception in java is the occurrence during computation that is anomalous and is not expected.   Exception handling is the mechanism which is used to handle such situations |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q27. What is an exception and exception handling in Java ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2017-01-09 13:01:40   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. An Exception in java is the occurrence during computation that is anomalous and is not expected.   Exception handling is the mechanism which is used to handle such situations. |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Q29. Write a program to swap two variables without using third** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | 2017-01-19 15:38:34   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. public static void main(String[] args) {    int num1 = 1;       int num2 = 2;    num1 = num1^num2;    num2 = num1^num2;    num1 = num1^num2;    System.out.print("num1 = " + num1 +", num2 = "+num2); } |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **30. What are the components of Spring framework ?** | [Spring](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Spring) | 2017-02-06 11:57:01   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. Core: [Core, Bean, Context, Expression Language]  Web: [Web, Portlet, Servlet, etc]  Data Access [JMS, JDBC, etc]  AOP, Aspect   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **If your application has multiple databases, how many changes you have to made in Hibernate?** | [Hibernate](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Hibernate) | 2018-07-04 16:21:07   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | | | Ans. If you have mulitple Databases you just need to add multiple configuration files and create the corresponding SessionFactory object. | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **38. If SOAP is more secure than Rest, Why many applications today use Rest ?** | [Web Service](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Web%20Service) | 2018-07-04 16:21:38   |  |  | | --- | --- | | http://javasearch.buggybread.com/InterviewQuestions/author.png |  | | |
| Ans. SOAP supports only xml whereas Rest support many other formats.  Rest services are very light weight compared to SOAP.  SOAP services are code heavy as they require creation. of stubs and skeltons. |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **40. How to sort objects based on one of the field ?** | [Core Java](https://javasearch.buggybread.com/InterviewQuestions/questionSearch.php?searchOption=label&keyword=Core%20Java) | |  |  | | --- | --- | |  |  | | |
| Ans. Using comparable and comparator and sorted collections like TreeSet or TreeMap.  or  use stream api from java 8 onwards which internally refers to comparable and comparator through lambda expressions |