Rico-Kali Hayes
Java SDET
07/23/2023

Core Java Interview Questions

1. **What is difference between JRE, JDK and JVM?**
   The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

   JRE stands for “Java Runtime Environment” and may also be written as “Java RTE.” The Java Runtime Environment provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

   Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language.

2. **What does static keyword mean in JAVA?**
   Static keyword Indicates that the particular member belongs to a type itself, rather than to an instance of that type. This means that only one instance of that static member is created which is shared across all instances of the class.

3. **What is Object and Class in JAVA?**
   - Class: A Class is a template for creating an object. It contains parameters and methods.
   - An Object is an instantiation of a class. Java objects have three primary characteristics: identity, state, and behavior.
   -

4. **What are Immutable objects in JAVA?**
   An object is immutable when its state doesn't change after it has been initialized. For example, String is an immutable class and, once instantiated, the value of a String object never changes.

5. **What is “this” keyword in JAVA?**
   This is a reference variable that refers to the current class instance variable. It can be used to invoke current class methods, and current class constructors.

6. **What is Constructor in JAVA?**
   A Constructor in java is used to create the instance of the class (instantiate). Constructors are special methods and its name is the same as the class name and it has no return type.

7. **What is private constructor?**
   A private constructor in Java is used in restricting object creation. It is a special instance constructor used in static member-only classes. If a constructor is declared as private, then its objects are only accessible from within the declared class.

8. **Can you inherit constructor from another class?**
   No, constructors cannot be inherited in Java. In inheritance sub class inherits the members of a super class except constructors. In other words, constructors cannot be inherited in Java therefore, there is no need to write final before constructors.

9. **What are different OOPS Concepts?**
   Abstraction, encapsulation, inheritance, and polymorphism

10. **What are different access modifiers in JAVA?**
    Private, Public, Default, and Protected

11. **How do you achieve inheritance?**
    To inherit the parent class, a child class must include a keyword called "extends." The keyword "extends" enables the compiler to understand that the child class derives the functionalities and members of its parent class.

12. **How do you refer to parent class objects in JAVA?**
    By using the super() keyword, which calls the constructor of the parent class.

13. **What class is superclass of all classes in JAVA?**
    Object class is the root or superclass of the class hierarchy.

14. **Is multiple inheritance supported in JAVA?**
    No it is not supported in Java.

15. **Why it's not supported? How can you achieve multiple inheritance in JAVA?**
    The reason behind this is to prevent ambiguity. Consider a case where class B extends class A and Class C and both class A and C have the same method display(). Now java compiler cannot decide, which display method it should inherit.

16. **How do you achieve Encapsulation?**
    - Declaring the variables of a class as private.
    - Providing public setter and getter methods to modify and view the variables values.

17. **What is Polymorphism?**
    Polymorphism means many forms. There are two types of polymorphism which are Run-time (Method Overriding) and Compile time (Method Overloading) polymorphism.

18. **What is difference between Overriding and OverLoading?**
    Method Overriding is when a child class has the same method of the parent class through inheritance. The method has the same return type and parameters as the Parent class, but the method implementation is different. Method Overloading is when a class has multiple methods with the same name, but must have either different return type, different data type of the parameters in the method signature, or different number of parameters in the method signature.

19. **Can we overload a constructor?**
    Yes, we can overload a constructor.

20. **Can we override static methods in JAVA? What happens if you do? Runtime or compile time error?**
    No, we can't override static methods since method overriding relies on dynamic binding at runtime, but static methods are bonded at compile time with static binding. As a result, we are unable to override static methods.

21. **Can we override private methods in JAVA?**
    No, we cannot override private or static methods in Java. Private methods in Java are not visible to any other class which limits their scope to the class in which they are declared.

22. **What is Abstraction?**
23. Abstraction in Java is a process of hiding the implementation details from the user and showing only the functionality to the user. It can be achieved by using abstract classes, methods, and interfaces.
24.
25. **Can you make an object out of an Abstract class? How?**
    An abstract class is a class that cannot be instantiated and is meant to be inherited by concrete classes.

26. **Can there be an abstract method without an abstract class?**
An abstract class is not required to have an abstract method in it. But any class that has an abstract method in it or that does not provide an implementation for any abstract methods declared in its superclasses must be declared as an abstract class.

27. **What are interfaces in JAVA?**
An interface in Java is a blueprint of a behaviour. A Java interface contains static constants and abstract methods. In its most common form, an interface is a group of related methods with empty bodies.

28. **How do you create an interface?**
An interface is declared by using the interface keyword.

29. **What is aggregation in JAVA?**
When an object A contains a reference to another object B or we can say Object A has a HAS-A relationship with Object B, then it is termed as Aggregation. Aggregation helps in reusing the code.

30. **What is Composition in JAVA?**
It occurs when a class references one or more objects of other classes in a single instance. For example, a Class Car is a composition of class Engine and class wheels and a Class Body is a composition of the class Heart, class Stomach, etc.

31. **Difference between aggregation and composition in JAVA?**
In Aggregation, linked objects are independent of each other. In Composition, objects are tightly coupled or dependent on each other.

32. **What is exception handling in JAVA?**
Exception handling is the process of responding to unwanted or unexpected events when a computer program runs. Exception handling deals with these events to avoid the program or system crashing, and without this process, exceptions would disrupt the normal operation of a program.

33. **What is the difference between checked and unchecked exception?**
Checked Exceptions are the exceptions that are checked at compile time. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using the throws keyword. Some common examples of Checked Exceptions include IOException, ParseException, and ClassNotFoundException. Unchecked Exceptions are the exceptions that are not checked at compile time. Exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under Throwable class is checked. Examples of Unchecked Exceptions include NullPointerException, ArrayIndexOutOfBoundsException, and IllegalArgumentException.

34. **Give me example of some exception that you have faced?**
Some examples of exceptions I faced are: ArrayIndexOutOfBoundsException which is an unchecked exception that happens when the program is trying to access an invalid array index that does not exist. I also faced a FileNotFoundException, which is a checked exception that occurs when the program is trying to access a file that does not exist in the defined location.

35. **What is difference between throw and throws in exception?**
The throws keyword is used to declare which exceptions can be thrown from a method, while the throw keyword is used to explicitly throw an exception within a method or block of code.

36. **What is the difference between final, finally and finalize?**
The Final keyword is a non-access modifier used for classes, attributes and methods, which makes them non-changeable. Finally is used as a block to execute a set of statements after a Try or Catch Block. Finalize is used as a method to perform cleanup operations.

37. **Can we have more than one catch in try-catch statement?**
Yes, there can be multiple catch blocks.

38. **Can we have more than one finally in try-catch statement?**
We can only have one finally block after a try catch statement.

39. **Can we have finally block without catch block?**
Yes, there can be a finally block after a try block without the use of a catch block.

40. **What happens if you use a return statement or a system.exit(0) call in a try block, does the finally block get executed?**
Yes, the finally block will be executed even after a return statement in a method. The finally block will always execute even an exception occurred or not in Java. If we call the System. exit() method explicitly in the finally block then only it will not be executed.

41. **Is it possible to create custom exceptions? how do you do it?**
Yes we can create custom exceptions. We do this by having the custom exception class extend the Exception class that belongs to java.lang package.

42. **What are collections in JAVA?**
Any group of individual objects which are represented as a single unit is known as a collection of objects.

43. **What is difference between ArrayList and LinkedList?**
    - ArrayList internally uses a dynamic array to store the elements. Manipulation with ArrayList is slow because it internally uses an array. If any element is removed from the array, all the other elements are shifted in memory. An ArrayList class can act as a list only because it implements List only. ArrayList is better for storing and accessing data. The memory location for the elements of an ArrayList is contiguous.
    - LinkedList internally uses a doubly linked list to store the elements. Manipulation with LinkedList is faster than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory. LinkedList class can act as a list and queue both because it implements List and Deque interfaces. LinkedList is better for manipulating data. The location for the elements of a linked list is not contagious.
    -

44. **Difference between TreeSet and TreeMap?**
    - TreeSet stores unique elements. Elements are stored in a sorted order based on their natural ordering. It does not allow null elements and implements the Set interface.
    - The TreeMap stoers key value paris, where each key is unique. Elements are stored in a sorted order based on the keys natural ordering. It allows null values, but not null keys and implements the Map interface.
    -

45. **Difference between stack and heap?**
The major difference between Stack memory and heap memory is that the stack is used to store the order of method execution and local variables, while the heap memory stores the objects and it uses dynamic memory allocation and deallocation.

46. **What is stack memory and heap memory? Are they allocated by each thread or JVM instance?**
    - Stack: Stack memory is used for the execution of a thread. They contain method-specific values that are short-lived and references to other objects in the heap that is getting referred from the method. Stack memory is always referenced in LIFO (Last-In-First-Out) order.
    - Heap: Heap is the area of memory used to store objects instantiated by applications running on the JVM. When the JVM is started, heap memory is created and any objects in the heap can be shared between threads as long as the application is running.
    -

47. **What is HashSet? vs TreeSet?**
    - HashSet: A HashSet in Java is a class from the Collections Framework. It allows you to store multiple unique values in a collection using a hash table. Its internal structure is optimized for faster searches
    - TreeSet: A TreeSet provides an implementation of the Set interface that uses a tree for storage. Objects are stored in a sorted and ascending order. Access and retrieval times are fast, and makes TreeSet an excellent choice when storing large amounts of sorted information that must be found quickly.
    -

48. **What is HashMap? vs LinkedHashMap?**
    - HashMap: The HashMap class of the Java collections framework provides the functionality of the hash table data structure. It stores elements in key/value pairs. Here, keys are unique identifiers used to associate each value on a map. The HashMap class implements the Map interface.
    - LinkedHashMap: The LinkedHashMap class of the Java collections framework provides the hash table and linked list implementation of the Map interface. The LinkedHashMap interface extends the HashMap class to store its entries in a hash table. It internally maintains a doubly-linked list among all of its entries to order its entries.
    -

49. **What is Concurrency in JAVA?**
    Java Concurrency is the capability of the Java platform to run multiple operations simultaneously. The operations could be multiple Java programs or parts of a single Java program. Java Concurrency relies on two essential components such as threads and processes.

50. **What is Thread is JAVA?**
    A thread in Java is the direction or path that is taken while a program is being executed. Generally, all the programs have at least one thread, known as the main thread, that is provided by the JVM or Java Virtual Machine at the starting of the program's execution.

51. **How do you start a Thread in JAVA?**
    You need to start each thread manually by calling the start() method of the Thread class. This method first creates a thread and then calls the run() method of the Runnable task you have passed to this new thread.

52. **What is Runnable interface in JAVA?**
    A runnable interface in Java is an interface whose instances can run as a Thread.

53. **How do you prevent deadlock in threads?**
    Deadlock can be avoided by using synchronization, avoiding nested locks, avoiding unnecessary locks, and by using Thread.join().

54. **What is the difference between a Feature and Story?**
    Stories are small parts of a feature that allow teams to put context to their actions. Each completed user story iteratively builds the feature. Only when all of the user stories have been completed can you consider a feature deliverable.

55. **What are JAVA Reflections API?**
Java reflection is an API used to inspect and modify Java classes, fields, methods, and constructors at runtime. Reflection can be used to get information about a class, constructor, and methods.

56. **How to dynamically load classes in JAVA?**
The Java ClassLoader is a part of the Java Runtime Environment that dynamically loads Java classes into the Java Virtual Machine.

57. **How do you convert JSON object to Java object?**
We can convert a JSON to Java Object using the readValue() method of ObjectMapper class, this method deserializes a JSON content from given JSON content String.

58. **How will you convert character array into string?**
The valueOf() and copyValueOf() methods of String class can be used to convert a char array into a string.

59. **What is the contract between hashcode() and equals() in Java?**
hashCode() and equals() contract: The basic rule of the contract states that if two objects are equal to each other based on equals() method, then the hash code must be the same, but if the hash code is the same, then equals() can return false.