TestLeaf

Core Java Interview Questions

1. What is immutability in java?

Immutability means - once created, the objects state cannot be changed; an object which state is guaranteed to stay identical over its entire lifetime. Immutable objects are particularly useful in concurrent applications. Since they cannot change state, they cannot be corrupted by thread interference or observed in an inconsistent state.

2. Difference between Hashmap and Hashtable?

Features	HashMap	HashTable
Synchronization	non-synchronized	synchronized
Thread-safe	×	✓
Allow Null	✓	×
Speed on Single Thread	Faster	Slower in relative

3. Difference between Method Overloading and Overriding?

Features	Overloading	Overriding
Meaning	Same method name, return type but with different argument(s)	Same method signature in child overrides the parent class
When it occurs?	During compile	@ Runtime
Why?	Increase the readability of the program.	Customized implementation of the method that is already provided by its super class.
Where?	Within a class	2 classes that have IS-A relationship
Return type	Same	Can be different

4. Difference between interface and abstract class?

Features	Interface	Abstract Class
Meaning	Interfaces are rules (Rules because you must give an implementation to them and that you can't ignore or avoid, so that are imposed like rules) which works as a common understanding document among the various teams.	Abstract declarations are like rules to be followed and concrete implementations are like guidelines (You can use that as it is or you can ignore it by overriding and giving your own choice implementation to it).
Relationship	implement several interfaces.	extend only one abstract class
instance variables	Cannot have	Can have
Constructor	Cannot have	Can have
Speed	Slower in relative	Faster

5. Difference between List and ArrayList?

Features	List	ArrayList
What it is?	interface	implementation of List interface
Length	fixed length data structure	Dynamic (can resize)
Stores	both primitives and Objects in Java	not store primitives in ArrayList

6. Difference between int and Integer?

Features	int	Integer
What it is?	Primitive data type	A class with a single field of type – int
Example	int count = 2;	Integer count = 2; which is interpreted as Integer count = Integer.valueOf(2);

7. Difference between Set and List?

Features	Set	List
What it is?	Collection with no duplicates and unordered	ordered Collection and can allow duplicates
Example	getWindowHandles() returns unique unordered collection (Set)	findElements() returns ordered collection (List)
Null	Only one Null allowed	Multi Null allowed (as duplicated allowed)

8. What is the use of Super Keyword?

If your method overrides one of its superclass's methods, you can invoke the overridden method through the use of the keyword super.
You can also use super to refer to a hidden field (although hiding fields is discouraged)

9. Difference between Vector and ArrayList?

Features	Vector	ArrayList
Synchronization	synchronized	non-synchronized
Thread-safe	✓	×
Speed	Slower in relative	Faster
Capacity	grow by half of its size when resized	doubles the size of itself by default when grows

10. What will happen if you call return statement or System.exit on try or catch – finally block?

Finally block will execute even if you put return statement in try block or catch block but finally block won't run if you call System.exit form try or catch.

11. Can you override private or static method in Java?

You cannot override private or static method in Java, if you create similar method with same return type and same method arguments that's called method hiding.

12. Can you override private or static method in Java?

You cannot override private or static method in Java, if you create similar method with same return type and same method arguments that's called method hiding.

13. Difference between static and final?

Features	static	final
Scope	Variables, methods	Variables, methods, classes
How it works?	do not require an object	final variable cannot be reassigned. final method cannot be overridden. final class cannot be inherited.
Speed	Slower in relative	Faster

14. What is Composition in Java?

Creating an object of other class in your class and calling other class variables and methods is known as composition (known as has-a relationship as one class "has a" object of other class).

15. How to fetch only specific character from a given string like only numbers, special characters etc.?

This can be done either of the given ways:

a) Use Regular Expressions with predefined character classes. For example, here are the pattern API predefinitions in Java:

Construct	Description
.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\x0B\f\r]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

So, if you wish to have only numbers, then use this code
str.replaceAll("\\D+",""); // this will delete the non-digits

b) Use ASCII characters to find and remove the unexpected values from the given string. For example, the ascii values for 0 to 9 is 47 to 58. In that case, use the following code, to delete others

```
final StringBuilder sb = new StringBuilder(input.length())
for(int i = 0; i < input.length(); i++){
    final char c = input.charAt(i);
    if(c > 47 && c < 58){
        sb.append(c);
    }
}
return sb.toString();
```

16. How to fetch only specific character from a given string like only numbers, special characters etc.?

Features	String	StringBuffer	StringBuilder
Mutable	**Immutable** Once created, cannot change	**mutable** can change the value of the object	**mutable** can change the value of the object

Example	`String SA = new String ("Test");`
	`String SB = SA;`
	`SA = SA+"Leaf";`
	
	`System.out.println(SA);`
	`System.out.println(SB);`
	
	`// SA will print as – TestLeaf; whereas SB will print - Test`
		`StringBuffer SA = new StringBuffer("Test");`
		`StringBuffer SB = SA;`
		`SA = SA.append("Leaf");`
		
		`System.out.println(SA);`
		`System.out.println(SB);`
		
		`// Both SA and SB will print as - TestLeaf`
			`StringBuilder SA = new StringBuilder ("Test");`
			`StringBuilder SB = SA;`
			`SA = SA.append("Leaf");`
			
			`System.out.println(SA);`
			`System.out.println(SB);`
			
			`// Both SA and SB will print as - TestLeaf`

Safety	Thread-Safe Cannot be used by two threads simultaneously.	Thread-Safe Synchronized	Not Thread-Safe Not Synchronized
Performance	Fast	Very slow	Fast

17. How to get the decimal part of a float or double?

You can use Math class floor method or in case of double - convert to int to deduct from the original value;

```
double d = 12345.4566; // option 1
double v = d - (int) d;
System.out.println(v);

Float f = 12345.4566f; // option 2
v = f - Math.floor(f);
System.out.println(v);

String s = ""+12345.4566; // option 3 (if you need just the decimals)
int index = s.indexOf('.');
System.out.println(Integer.parseInt(s.substring(index + 1)));
```

18. Difference between extends and implements?

Features	extends	implements
When?	***extends for extending a class.*** Inheritance between concrete (non-abstract) and abstract classes use extends keyword.	***implements for implementing an interface*** Inheritance between classes (including abstract classes) and interfaces, use implements keyword.
How many?	Extend only one class to another, any number of classes in a ladder; and this is known as **multilevel inheritance**. You cannot extend multiple classes to one class and this is known as **multiple inheritances** which Java does not support through classes.	After **extends** there should be only one class (either concrete or abstract) and after implements there can be any number of interfaces.

19. What is "this" keyword in Java?

Within an instance method or a constructor, **this** is a reference to the current object – the object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or a constructor by using **this**.

```
public class Point {
    public int x = 0;
    public int y = 0;

    //constructor
    public Point(int x, int y) {
        this.x = x; // this points to the local variable and not argument variable
        this.y = y;
    }
}
```

20. How to make the given string as palindrome?

This program assist to know given string is a palindrome; same can be used for numbers as well

```
     int length = str.length(); // find the length

      for ( int i = length - 1; i >= 0; i-- ) // reverse the string
         String reverse = reverse + original.charAt(i);

      if (str.equals(reverse)) // check if reversed matches the original
         System.out.println("Entered string is a palindrome.");
      else
         System.out.println("Entered string is not a palindrome.");
```

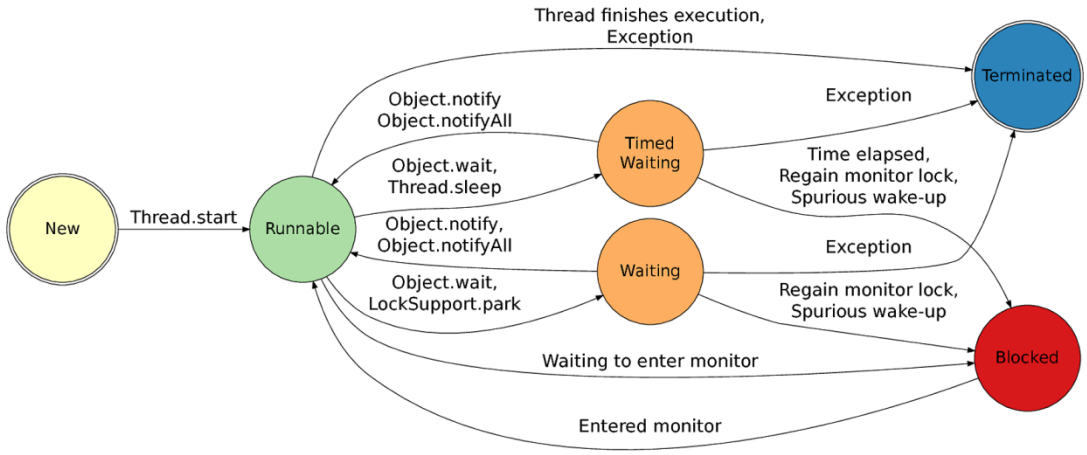
21. How to remove duplicate values from a given integer array?

This program assist to know given string is a palindrome; same can be used for numbers as well

```
return new HashSet<Integer>(Arrays.asList(arr)).toArray(new Integer[0]);

// asList -> Returns a fixed-size list backed by the specified array
// toArray-> Returns an array containing all of the elements in this list in
             proper sequence (from first to last element)
// HashSet-> implements the Set interface, backed by a hash table (so it has
             only unique values)
```

22. Explain how thread lifecycle works?



23. Difference between wait and sleep?

Features	Wait	sleep
Class	java.lang.Object	java.lang.Thread
when Thread is waiting	Releases the lock	Holds the lock
Method Type	Instance	Static
Can awake?	Yes – using notify() or notifyAll() method	No, cannot
Condition based?	Yes	No

24. What is the difference between JDK and JVM?

Java Development Kit (JDK) is for development purpose and JVM is a part of it to execute the java programs.

JDK provides all the tools, executables and binaries required to compile, debug and execute a Java Program. The execution part is handled by JVM to provide machine independence.

25. What is the difference between JVM and JRE?

Java Runtime Environment (JRE) is the implementation of JVM. JRE consists of JVM and java binaries and other classes to execute any program successfully. JRE doesn't contain any development tools like java compiler, debugger etc. If you want to execute any java program, you should have JRE installed.

26. Which class is the superclass of all classes?

java.lang.Object is the root class for all the java classes and we don't need to extend it.

27. Why Java doesn't support multiple inheritance?

Java doesn't support multiple inheritance in classes because of "Diamond Problem". However multiple inheritance is supported in interfaces. An interface can extend multiple interfaces because they just declare the methods and implementation will be present in the implementing class. So there is no issue of diamond problem with interfaces.

28. Why Java is not pure Object Oriented language?

Java is not said to be pure object oriented because it support primitive types such as int, byte, short, long etc. I believe it brings simplicity to the language while writing our code. Obviously java could have wrapper objects for the primitive types but just for the representation, they would not have provided any benefit.

As we know, for all the primitive types we have wrapper classes such as Integer, Long etc that provides some additional methods.

29. What is difference between path and classpath variables?

PATH is an environment variable used by operating system to locate the executables. That's why when we install Java or want any executable to be found by OS, we need to add the directory location in the PATH variable.

Classpath is specific to java and used by java executables to locate class files. We can provide the classpath location while running java application and it can be a directory, ZIP files, JAR files etc.

30. What is the importance of main method in Java?

main() method is the entry point of any standalone java application. The syntax of main method ispublic static void main(String args[]).
main method is public and static so that java can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program.

31. Can we overload main method?

Yes, we can have multiple methods with name "main" in a single class. However if we run the class, java runtime environment will look for main method with syntax as public static void main(String args[]).

32. Can we have multiple public classes in a java source file?

We can't have more than one public class in a single java source file. A single source file can have multiple classes that are not public.

33. What is Java Package and which package is imported by default?

Java package is the mechanism to organize the java classes by grouping them. The grouping logic can be based on functionality or modules based. A java class fully classified name contains package and class name. For example, java.lang.Object is the fully classified name of Object class that is part ofjava.lang package.
java.lang package is imported by default and we don't need to import any class from this package explicitly.

34. What are access modifiers?

Java provides access control through public, private and protected access modifier keywords. When none of these are used, it's called default access modifier.
A java class can only have public or default access modifier.

35. What is finally and finalize in java?

finally block is used with try-catch to put the code that you want to get executed always, even if any exception is thrown by the try-catch block. finally block is mostly used to release resources created in the try block.

finalize() is a special method in Object class that we can override in our classes. This method get's called by garbage collector when the object is getting garbage collected. This method is usually overridden to release system resources when object is garbage collected.

36. What is inner class in java?

We can define a class inside a class and they are called nested classes. Any non-static nested class is known as inner class. Inner classes are associated with the object of the class and they can access all the variables and methods of the outer class. Since inner classes are associated with instance, we can't have any static variables in them. We can have local inner class or anonymous inner class inside a class.

37. What is anonymous inner class?

A local inner class without name is known as anonymous inner class. An anonymous class is defined and instantiated in a single statement. Anonymous inner class always extend a class or implement an interface.

Since an anonymous class has no name, it is not possible to define a constructor for an anonymous class. Anonymous inner classes are accessible only at the point where it is defined.

38. Can we declare a class as static?

We can't declare a top-level class as static however an inner class can be declared as static. If inner class is declared as static, it's called static nested class. Static nested class is same as any other top-level class and is nested for only packaging convenience.

39. What is static import?

If we have to use any static variable or method from other class, usually we import the class and then use the method/variable with class name.

```
1 import java.lang.Math;
2
3 //inside class
4 double test = Math.PI * 5;
```

We can do the same thing by importing the static method or variable only and then use it in the class as if it belongs to it.

```
1 import static java.lang.Math.PI;
2
3 //no need to refer the class now
4 double test = PI * 5;
```

Use of static import can cause confusion, so it's better to avoid it. Overuse of static import can make your program unreadable and unmaintainable.

40. What is try-with-resources in java?

One of the Java 7 features is try-with-resources statement for automatic resource management. Before Java 7, there was no auto resource management and we should explicitly close the resource. Usually, it was done in the finally block of a try-catch statement. This approach used to cause memory leaks when we forgot to close the resource.

41. What is multi-catch block in java?

Java 7 one of the improvement was multi-catch block where we can catch multiple exceptions in a single catch block. This makes are code shorter and cleaner when every catch block has similar code.

If a catch block handles multiple exception, you can separate them using a pipe (|) and in this case exception parameter (ex) is final, so you can't change it.

42. What is an interface?

Interfaces are core part of java programming language and used a lot not only in JDK but also java design patterns, most of the frameworks and tools. Interfaces provide a way to achieve abstraction in java and used to define the contract for the subclasses to implement.

Interfaces are good for starting point to define Type and create top level hierarchy in our code. Since a java class can implements multiple interfaces, it's better to use interfaces as super class in most of the cases.

43. What is an abstract class?

Abstract classes are used in java to create a class with some default method implementation for subclasses. An abstract class can have abstract method without body and it can have methods with implementation also.

abstract keyword is used to create a abstract class. Abstract classes can't be instantiated and mostly used to provide base for sub-classes to extend and implement the abstract methods and override or use the implemented methods in abstract class.

44. Can an interface implement or extend another interface?

Interfaces don't implement another interface, they extend it. Since interfaces can't have method implementations, there is no issue of diamond problem. That's why we have multiple inheritance in interfaces i.e an interface can extend multiple interfaces.

45. What is Marker interface?

A marker interface is an empty interface without any method but used to force some functionality in implementing classes by Java. Some of the well known marker interfaces are Serializable and Cloneable.

46. What is static block?

Java static block is the group of statements that gets executed when the class is loaded into memory by Java ClassLoader. It is used to initialize static variables of the class. Mostly it's used to create static resources when class is loaded.

47. What are Wrapper classes?

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final. Java 5 autoboxing and unboxing allows easy conversion between primitive types and their corresponding wrapper classes.

48. What is Enum in Java?

Enum was introduced in Java 1.5 as a new type whose fields consist of fixed set of constants. For example, in Java we can create Direction as enum with fixed fields as EAST, WEST, NORTH, SOUTH. Enum constants are implicitly static and final.

49. What is Java Annotations?

Java Annotations provide information about the code and they have no direct effect on the code they annotate. Annotations are introduced in Java 5. Annotation is metadata about the program embedded in the program itself. It can be parsed by the annotation parsing tool or by compiler. We can also specify annotation availability to either compile time only or till runtime also. Java Built-in annotations are **@Override, @Deprecated and @SuppressWarnings.**

50. What is Java Reflection API? Why it's so important to have?

Java Reflection API provides ability to inspect and modify the runtime behavior of java application. We can inspect a java class, interface, enum and get their methods and field details. Reflection API is an advanced topic and we should avoid it in normal programming. Reflection API usage can break the design pattern such as Singleton pattern by invoking the private constructor i.e violating the rules of access modifiers.

51. What is the benefit of Composition over Inheritance?

One of the best practices of java programming is to "favor composition over inheritance". Some of the possible reasons are:

- Any change in the superclass might affect subclass even though we might not be using the superclass methods. For example, if we have a method test() in subclass and suddenly somebody introduces a method test() in superclass, we will get compilation errors in subclass. Composition will never face this issue because we are using only what methods we need.
- Inheritance exposes all the super class methods and variables to client and if we have no control in designing superclass, it can lead to security holes. Composition allows us to provide restricted access to the methods and hence more secure.
- We can get runtime binding in composition where inheritance binds the classes at compile time. So composition provides flexibility in invocation of methods.

52. What is Classloader in Java?

Java Classloader is the program that loads byte code program into memory when we want to access any class. We can create our own classloader by extending ClassLoader class and overriding loadClass(String name) method.

53. What are different types of classloaders?

There are three types of built-in Class Loaders in Java:

D. Bootstrap Class Loader – It loads JDK internal classes, typically loads rt.jar and other core classes.

E. Extensions Class Loader – It loads classes from the JDK extensions directory, usually $JAVA_HOME/lib/ext directory.

F. System Class Loader – It loads classes from the current classpath that can be set while invoking a program using -cp or -classpath command line options.

54. What is ternary operator in java?

Java ternary operator is the only conditional operator that takes three operands. It's a one liner replacement for if-then-else statement and used a lot in java programming. We can use ternary operator if-else conditions or even switch conditions using nested ternary operators.

55. How to sort a collection of custom Objects in Java?

We need to implement Comparable interface to support sorting of custom objects in a collection. Comparable interface has compareTo(T obj) method which is used by sorting methods and by providing this method implementation, we can provide default way to sort custom objects collection.

However, if you want to sort based on different criteria, such as sorting Employees collection based on salary or age, and then we can create Comparator instances and pass it as sorting methodology.

56. What is break and continue statement?

We can use break statement to terminate for, while, or do-while loop. We can use break statement in switch statement to exit the switch case. You can see the example of break statement at java break. We can use break with label to terminate the nested loops.

The continue statement skips the current iteration of a for, while or do-while loop. We can use continue statement with label to skip the current iteration of outermost loop.

57. What is default constructor?

No argument constructor of a class is known as default constructor. When we don't define any constructor for the class, java compiler automatically creates the default no-args constructor for the class. If there are other constructors defined, then compiler won't create default constructor for us.

58. Can we have try without catch block?

Yes, we can have try-finally statement and hence avoiding catch block.

59. What is Garbage Collection?

Garbage Collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects.In Java, process of deallocating memory is handled automatically by the garbage collector.

We can run the garbage collector with code Runtime.getRuntime().gc() or use utility method System.gc()

60. What is instanceof keyword?

We can use instanceof keyword to check if an object belongs to a class or not. We should avoid it's usage as much as possible.

61. Can we use String with switch case?

One of the Java 7 feature was improvement of switch case of allow Strings. So if you are using Java 7 or higher version, you can use String in switch-case statements.

62. What is Serialization and Deserialization?

We can convert a Java object to an Stream that is called Serialization. Once an object is converted to Stream, it can be saved to file or send over the network or used in socket connections.

The object should implement Serializable interface and we can use java.io.ObjectOutputStream to write object to file or to any OutputStream object.

The process of converting stream data created through serialization to Object is called deserialization.

63. What is the use of System class?

Java System Class is one of the core classes. One of the easiest way to log information for debugging is System.out.print() method.

System class is final so that we can't subclass and override it's behavior through inheritance. System class doesn't provide any public constructors, so we can't instantiate this class and that's why all of it's methods are static.

Some of the utility methods of System class are for array copy, get current time, reading environment variables.

64. Java is Pass by Value or Pass by Reference?

You know that object variables contain reference to the Objects in heap space. When you invoke any method, a copy of these variables is passed and gets stored in the stack memory of the method. Hence it is "pass by value".

65. What is difference between Heap and Stack Memory?

Major difference between Heap and Stack memory are as follows:

- Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.
- Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it. Stack memory only contains local primitive variables and reference variables to objects in heap space.
- Memory management in stack is done in LIFO manner whereas it's more complex in Heap memory because it's used globally.