Basic Questions

Q) Is Java platform independent?
Yes. Java is a platform independent language. We can write java code on one platform and run it on another platform. For e.g. we can write and compile the code on windows and can run it on Linux or any other supported platform. This is one of the main features of java.

Q) What all memory areas are allocated by JVM?
Heap, Stack, Program Counter Register and Native Method Stack.

Q) Java vs. C ++?
The following features of java make it different from the C++:

- Simple
- Multi-threaded
- Distributed Application
- Robust
- Security
- Complexities are removed (Pointers, Operator overloading).

Q) What is javac ?
It produces the java byte code from *.java file. It is the intermediate representation of your source code that contains instructions.

Q) What is class?
Class is nothing but a template that describes the data and behavior associated with instances of that class

Q) What is the base class of all classes?
java.lang.Object

Q) Path and ClassPath
Path specifies the location of .exe files while classpath is used for specifying the location of .class files.

Q) Different Data types in Java.

- byte – 8 bit (are esp. useful when working with a stream of data from a network or a file).
- short – 16 bit
- char – 16 bit Unicode
- int – 32 bit (whole number)
- float – 32 bit (real number)
- long – 64 bit (Single precision)

- double – 64 bit (double precision)

Note: Any time you have an integer expression involving bytes, shorts, ints and literal numbers, the entire expression is promoted to int before the calculation is done.

Q) What is Unicode?
Java uses Unicode to represent the characters. Unicode defines a fully international character set that can represent all of the characters found in human languages.

Q) What are Literals?
A literal is a value that may be assigned to a primitive or string variable or passed as an argument to a method.

Q) Dynamic Initialization?
Java allows variables to be initialized dynamically, using any expression valid at the time the variable is declared.

Q) What is Type casting in Java?
To create a conversion between two incompatible types, we must use a cast. There are two types of casting in java: automatic casting (done automatically) and explicit casting (done by programmer).

Q) Arrays?
An array is a group of fixed number of same type values.

Q) What is break statement in java?
It is also referred as terminator. In Java, the break statement can be used in following two cases:

- It terminates a statement sequence in a switch-case statement.
- It can be used to come out of a loop

Q) Why can't I do myArray.length () ? Arrays are just objects, right?
Yes, the specification says that arrays are object references just like classes are. You can even invoke the methods of Object such as toString () and hashCode () on an array. However, length is a data item of an array and not a method. So you have to use myArray.length.

Q) How can I put all my classes and resources into one file and run it?
Use a JAR file. Put all the files in a JAR, then run the app like this:

Java -jar [-options] jarfile [args...]

Q) Can I declare a data type inside loop in java?
Any Data type declaration should not be inside the loop.

Q) java.lang.* get imported by default. For using String and Exception classes, you don't need to explicitly import this package. The major classes inside this package are

- Object class
- Data type wrapper classes
- Math class
- String class
- System and Runtime classes
- Thread classes
- Exception classes
- Process classes
- Class classes

Q) Arrays can be defined in different ways. Write them down.

```
int arr[] = null;
int arr[][] = new int arr[][];
int [][] arr = new arr [][];
int [] arr [] = new arr[][];
```

OOPs Interview Questions

Q) Four main principles of OOPS language?

- Inheritance
- Polymorphism
- Data Encapsulation
- Abstraction

Q) What is inheritance?
The process by which one class acquires the properties and functionalities of another class. Inheritance brings reusability of code in a java application.

Q) What is Polymorphism and what are the types of it?
Polymorphism is the ability of an object to take many forms. The most common use of polymorphism in OOPs is to have more than one method with the same name in a single class.

Q) What is the method overriding?
It is a feature using which a child class overrides the method of parent class. It is only applicable when the method in child class has the signature same as parent class.

Q) Can we override a static method?
No, we cannot override a static method.

Q) What is method overloading?
Having more than one method with the same name but different number, sequence or types of arguments is known is method overloading.

Q) Does Java support operator overloading?
Operator overloading is not supported in Java.

Q) Can we overload a method by just changing the return type and without changing the signature of method?
No, We cannot do this.

Q) What is static and dynamic binding?
Binding refers to the linking of method call to its body. A binding that happens at compile time is known as static binding while binding at runtime is known as dynamic binding.

Q) What is Encapsulation?
Encapsulation means the localization of the information or knowledge within an object.
Encapsulation is also called as “Information Hiding”.

Q) Abstract class?
An abstract class is a class which can’t be instantiated (we cannot create the object of abstract class), we can only extend such classes. It provides the generalized form that will be shared by all of its subclasses, leaving it to each subclass to fill in the details. We can achieve partial abstraction using abstract classes, to achieve full abstraction we use interfaces.

Q) What is Interface in java?
An interface is a collection of abstract, static and default methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Q) What is the difference between abstract class and interface?
With abstract classes, you can declare fields that are not static and final, and define public, protected, and private concrete methods.

With interfaces, all fields are automatically public, static, and final, and all methods that you declare or define (as default methods) are public.

In addition, you can extend only one class, whether or not it is abstract, whereas you can implement any number of interfaces.

Q) Which access modifiers can be applied to the inner classes?
public, private , abstract, final, protected.

Q) What are Constructors?
Constructors are used for creating an instance of a class, they are invoked when an instance of class gets created. Constructor name and class name should be same and it doesn't have a return type.

Q) Can we inherit the constructors?
No, we cannot inherit constructors.

Q) Can we mark constructors final?
No, Constructor cannot be declared final.

Q) What is default and parameterized constructors?
Default: Constructors with no arguments are known as default constructors, when you don't declare any constructor in a class, compiler creates a default one automatically.

Parameterized: Constructor with arguments are known as parameterized constructors.

Q) Can a constructor call another constructor?
Yes. A constructor can call another constructor of the same class using this keyword. For e.g. this() calls the default constructor.
Note: this() must be the first statement in the calling constructor.

Q) Can a constructor call the constructor of parent class?
Yes. In fact it happens by default. A child class constructor always calls the parent class constructor. However we can still call it using super keyword. For e.g. super() can be used for calling super class default constructor.

Note: super() must be the first statement in a constructor.

Q) this keyword?
The this keyword is a reference to the current object.

Q) Can this keyword be assigned null value?
No, this keyword cannot have null values assigned to it.

Q) Explain ways to pass the arguments in Java?
In java, arguments can be passed in 2 ways,

Pass by value – Changes made to the parameter of the subroutines have no effect on the argument used to call it.
Pass by reference – Changes made to the parameter will affect the argument used to call the subroutine.

Q) What is static variable in java?
Static variables are also known as class level variables. A static variable is same for all the objects of that particular class in which it is declared.

Q) What is static block?
A static block gets executed at the time of class loading. They are used for initializing static variables.

Q) What is a static method?
Static methods can be called directly without creating the instance (Object) of the class. A static method can access all the static variables of a class directly but it cannot access non-static variables without creating instance of class.

Q) Explain super keyword in Java?
super keyword references to the parent class. There are several uses of super keyword:

- It can be used to call the superclass(Parent class) constructor.
- It can be used to access a method of the superclass that has been hidden by subclass (Calling parent class version, In case of method overriding).

Q) Use of final keyword in Java?
Final methods – These methods cannot be overridden by any other method.
Final variable – Constants, the value of these variable can't be changed, it's fixed.
Final class – Such classes cannot be inherited by other classes. These type of classes will be used when application required security or someone don't want that particular class.

Q) What is a Object class?
This is a special class defined by java; all other classes are subclasses of object class. Object class is superclass of all other classes. Object class has the following methods

- objectClone () – to creates a new object that is same as the object being cloned.
- boolean equals(Object obj) – determines whether one object is equal to another.
- finalize() – Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the finalize method to dispose of system resources or to perform other cleanup.
- toString () – Returns a string representation of the object.

Q) What are Packages?
A Package can be defined as a grouping of related types (classes, interfaces, enumerations and annotations)

Q) What is the difference between import java.util.Date and java.util.* ?
The star form (java.util.*) includes all the classes of that package and that may increase the compilation time – especially if you import several packages. However it doesn't have any effect run-time performance.

Q) What is static import?
The *static import* statement gives you a way to import the constants and static methods that you want to use so that you do not need to prefix the name of their class.

Q) Garbage collection in java?
Since objects are dynamically allocated by using the new operator, java handles the de-allocation of the memory automatically when no references to an object exist for a long time is called garbage collection. The whole purpose of Garbage collection is efficient memory management.

Q) Use of finalize() method in java?
finalize() method is used to free the allocated resource.

Q) How many times does the garbage collector calls the finalize() method for an object?
The garbage collector calls the finalize() method Only once for an object.

Q) What are two different ways to call garbage collector?
System.gc() OR Runtime.getRuntime().gc().

Q) Can the Garbage Collection be forced by any means?
No, its not possible. You cannot force garbage collection. You can call system.gc() methods for garbage collection but it does not guarantee that garbage collection would be done.

Exception handling Interview Questions

Q) What is an exception?
Exceptions are abnormal conditions that arise during execution of the program. It may occur due to wrong user input or wrong logic written by programmer.

Q) Exceptions are defined in which java package? OR which package has definitions for all the exception classes?
Java.lang.Exception
This package contains definitions for Exceptions.

Q) What are the types of exceptions?
There are two types of exceptions: checked and unchecked exceptions.
Checked exceptions: These exceptions must be handled by programmer otherwise the program would throw a compilation error.
Unchecked exceptions: It is up to the programmer to write the code in such a way to avoid unchecked exceptions. You would not get a compilation error if you do not handle these exceptions. These exceptions occur at runtime.

Q) What is the difference between Error and Exception?
Error: Mostly a system issue. It always occur at run time and must be resolved in order to proceed further.
Exception: Mostly an input data issue or wrong logic in code. Can occur at compile time or run time.

Q) What is throw keyword in exception handling?
The throw keyword is used for throwing user defined or pre-defined exception.

Q) What is throws keyword?
If a method does not handle a checked exception, the method must declare it using the throws keyword. The throws keyword appears at the end of a method's signature.

Q) Difference between throw and throws in Java
1.Throws clause in used to declare an exception and **thow** keyword is used to throw an exception explicitly.

2. **throw** is followed by an instance variable and **throws** is followed by exception class names.

3. The keyword **throw** is used inside method body to invoke an exception and **throws clause** is used in method declaration (signature).

for e.g.

Throw:

```
static{
try {
throw new Exception("Something went wrong!!");
} catch (Exception exp) {
System.out.println("Error: "+exp.getMessage());
}
}
```

Throws:

```
public void sample() throws ArithmeticException{
  //Statements
  //if (Condition : There is an error)
ArithmeticException exp = new ArithmeticException();
  throw exp;
...
}
```

4. By using **Throw keyword** in java you cannot throw more than one exception but using **throws** you can declare multiple exceptions.

for e.g.

Throw:

throw new ArithmeticException("An integer should not be divided by zero!!")
throw new IOException("Connection failed!!")

Throws:

throws IOException, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException

Q) Can static block throw exception?
Yes, A static block can throw exceptions. It has its own limitations: It can throw only Runtime exception (Unchecked exceptions), In order to throw checked exceptions you can use a try-catch block inside it.

Q) What is finally block?
Finally block is a block of code that always executes, whether an exception occurs or not. Finally block follows try block or try-catch block.

Q) ClassNotFoundException vs NoClassDefFoundError?
1) ClassNotFoundException occurs when loader could not find the required class in class path.
2) NoClassDefFoundError occurs when class is loaded in classpath, but one or more of the class which are required by other class, are removed or failed to load by compiler.

Q) Can we have a try block without catch or finally block?
No, we cannot have a try block without catch or finally block. We must have either one of them or both.

Q) Can we have multiple catch blocks following a single try block?
Yes we can have multiple catch blocks in order to handle more than one exception.

Q) Is it possible to have finally block without catch block?
Yes, we can have try block followed by finally block without even using catch blocks in between.

Q) When a finally block does not get executed?
The only time finally won't be called is if you call System.exit() or if the JVM crashes first.

Q) Can we handle more than one exception in a single catch block?
Yes we can do that using if-else statement but it is not considered as a good practice. We should have one catch block for one exception.

Java Multithreading Interview Questions

Q) What is Multithreading?
It is a process of executing two or more part of a program simultaneously. Each of these parts is known as threads. In short the process of executing multiple threads simultaneously is known as multithreading.

Q) What is the main purpose of having multithread environment?
Maximizing CPU usage and reducing CPU idle time

Q) What are the main differences between Process and thread? Explain in brief.
1) One process can have multiple threads. A thread is a smaller part of a process.
2) Every process has its own memory space, executable code and a unique process identifier (PID) while every thread has its own stack in Java but it uses process main memory and shares it with other threads.
3) Threads of same process can communicate with each other using keyword like wait and notify etc. This process is known as inter process communication.

Q) How can we create a thread in java?
There are two ways of creating a thread:
1) By Implementing Runnable interface.
2) By Extending Thread class.

Q) Explain yield and sleep?
yield() – It causes the currently executing thread object to temporarily pause and allow other threads to execute.

sleep() – It causes the current thread to suspend execution for a specified period. When a thread goes into sleep state it doesn't release the lock.

Q) What is the difference between sleep() and wait()?
sleep() – It causes the current thread to suspend execution for a specified period. When a thread goes into sleep state it doesn't release the lock

wait() – It causes current thread to wait until either another thread invokes the notify() method or the notifyAll() method for this object, or a specified amount of time has elapsed.

Q) What is a daemon thread?
A daemon thread is a thread, that does not prevent the JVM from exiting when the

program finishes but the thread is still running. An example for a daemon thread is the garbage collection.

Q) What does join( ) method do?
if you use join() , it makes sure that as soon as a thread calls join, the current thread(yes, currently running thread) will not execute unless the thread you have called join is finished.

Q) Preemptive scheduling vs. time slicing?
1) The preemptive scheduling is prioritized. The highest priority process should always be the process that is currently utilized.
2) Time slicing means task executes for a defined slice/ period of time and then enter in the pool of ready state. The scheduler then determines which task execute next based on priority or other factor.

Q) Can we call run() method of a Thread class?
Yes, we can call run() method of a Thread class but then it will behave like a normal method. To actually execute it in a Thread, you should call Thread.start() method to start it.

Q) What is Starvation?
Starvation describes a situation where a thread is unable to gain regular access to shared resources and is unable to make progress. This happens when shared resources are made unavailable for long periods by “greedy” threads. For example, suppose an object provides a synchronized method that often takes a long time to return. If one thread invokes this method frequently, other threads that also need frequent synchronized access to the same object will often be blocked.

Q) What is deadlock?
Deadlock describes a situation where two or more threads are blocked forever, waiting for each other.

Serialization interview Questions

Q: What is Serialization and de-serialization?
Serialization is a process of converting an object and its attributes to the stream of bytes. De-serialization is recreating the object from stream of bytes; it is just a reverse process of serialization.

Q) Do we need to implement any method of Serializable interface to make an object serializable?
No. In order to make an object serializable we just need to implement the interface Serializable. We don’t need to implement any methods.

Q) What is a transient variable?
1) transient variables are not included in the process of serialization.
2) They are not the part of the object's serialized state.
3) Variables which we don't want to include in serialization are declared as transient.

String interview questions

Q) A string class is immutable or mutable?
String class is immutable that's the reason once its object gets created, it cannot be changed further.

Q) Difference between StringBuffer and StringBuilder class?
1) StringBuffer is thread-safe but StringBuilder is not thread safe.
2) StringBuilder is faster than StringBuffer.
3) StringBuffer is synchronized whereas StringBuilder is not synchronized.

Q) What is toString() method in Java?
The toString() method returns the string representation of any object.

Java collections interview questions

Q) What is List?
Elements can be inserted or accessed by their position in the list, using a zero-based index.
A list may contain duplicate elements.

Q) What is Map?
Map interface maps unique keys to values. A key is an object that we use to retrieve a value later. A map cannot contain duplicate keys: Each key can map to at most one value.

Q) What is Set?
A Set is a Collection that cannot contain duplicate elements.

Q) Why ArrayList is better than Arrays?
Array can hold fixed number of elements. ArrayList can grow dynamically.

Q) What is the difference between ArrayList and LinkedList?
1) LinkedList store elements within a doubly-linked list data structure. ArrayList store elements within a dynamically resizing array.
2) LinkedList is preferred for add and update operations while ArrayList is a good choice for search operations.

Q) For addition and deletion. Which one is most preferred: ArrayList or LinkedList?
LinkedList. Because deleting or adding a node in LinkedList is faster than ArrayList.

Q) For searches. Which one is most preferred: ArrayList or LinkedList?
ArrayList. Searching an element is faster in ArrayList compared to LinkedList.

Q) What is the difference between ArrayList and Vector?
1) Vector is synchronized while ArrayList is not synchronized.
2) By default, Vector doubles the size of its array when it is re-sized internally. ArrayList increases by half of its size when it is re-sized.

Q) What is the difference between Iterator and ListIterator?
Following are the major differences between them:
1) Iterator can be used for traversing Set, List and Map. ListIterator can only be used for traversing a List.
2) We can traverse only in forward direction using Iterator. ListIterator can be used for traversing in both the directions(forward and backward).

Q) Difference between TreeSet and SortedSet?
TreeSet implements SortedSet interface.

Q) What is the difference between HashMap and Hashtable?
1) Hashtable is synchronized. HashMap is not synchronized.
2) Hashtable does not allow null keys or values. HashMap allows one null key and any number of null values.

Q) What is the difference between Iterator and Enumeration?
1) Iterator allows to remove elements from the underlying collection during the iteration using its remove() method. We cannot add/remove elements from a collection when using enumerator.
2) Iterator has improved method names.
Enumeration.hasMoreElement() -> Iterator.hasNext()
Enumeration.nextElement() -> Iterator.next().