JAVA INTERVIEW QUESTIONS

Most asked Interview Questions

BY : CODE OF GEEKS

CODE OF GEEKS

Java Interview Questions

This e-book contains **most frequently asked questions** on **Java – Programming language**.

Prepared by **CODE OF GEEKS**

In case of bug-reporting, please mail us at

support@codeofgeeks.com

SET 1 – Q1 – Q5

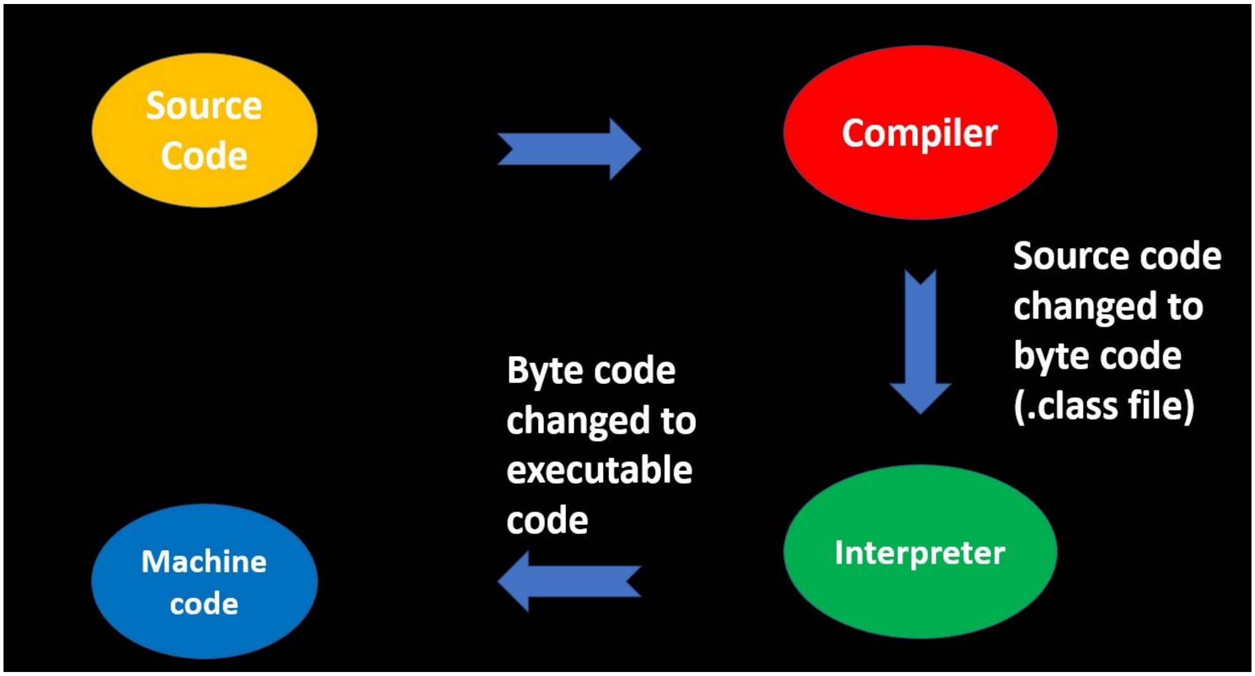
Q. What do you know about Java Programming Language ?
A. Java is a **general purpose**, **object oriented** programming language, designed by **James Gosling** in 1991. First version of Java was launched on **May 23, 1995**. Earlier, Java was named as **Oak**.

Q. Tell me some important features of Java.
A. Features of JAVA :

1. Java is class based **object oriented**, compiled & **Interpreted** programming language.
2. Platform Independent & **Portable**.
3. Java uses the concept of **Multithreading.**
4. Java is **dynamic** – as it can multiple programs and applications dynamically. Ex- Libraries.
5. Java is **Robust** & Secure.

Q. What are Identifiers in Java ?
A. Identifiers are the name given to a class, method, or a variable. Identifiers can start from alphabets, $ sign, but cannot start with numbers or any other sign, **except underscore**.

Q. Explain the procedure of compilation of a Java Code ?
A. For compiling a source code, Java uses both compiler as well as interpreter. Java program takes place at two stages : **Compiler & Interpreter**. Compiler changes a source code to byte code (.class file) and Interpreter changes byte code to machine code.Following is the procedure of compiling a Java Code :


Source Code
Compiler
Source code changed to byte code (.class file)
Byte code changed to executable code
Interpreter
Machine code

Q. What are access modifiers in Java ?
A. In Java, Access modifiers, specifies the accessibility or scope of a field, method, constructor, or a class.

SET 2 – Q6 – Q10

Q. Name some access modifiers in Java ?
A. There are **four access modifiers** in Java Programming Language :

1. public
2. protected
3. default.
4. private.

Q. Name some non access modifiers ?
A. ” **final, abstract, strictfp, synchronized, volatile** ” are non access modifiers.

Q. Explain me the role of JVM.
A. JVM stands for **Java Virtual Machine**. JVM is a software that provides run time environment in which java byte code can be executed.

Q. Explain JRE ?
A. JRE is **Java Runtime Environment**. It is the implementation of JVM.

Q. What is JDK ?
A. JDK is **Java Development kit**. It is the collection of programming tools, JRE, JVM.

SET 3 – Q11 – Q15

Q. What is Applet ?
A. It is a small application written in Java or any other language, that compiles in **Byte code**. It is basically a small internet based program embedded inside a pattern.

Q. What is a Class ?
A. A Class is a specification or a **template** of an object. Class contains variables and methods.

Q. What is an Object ?
A. Object is an instance (dynamic memory allocation) of a class. The process of creating objects out of class is known as instantiation. Object has a state, behaviour and an identity.

Q. Explain OOPS methodology ?
A. The entire OOPS methodology has been derived from a single root called object. In OOPS, all programs involve creation of classes and object. **Four important features** of OOPS are :

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

Q. What is Procedural oriented programming ?
A. It is the methodology where programming is done using procedures & functions.

SET 4 – Q16 – Q20

Q. Is Java an object oriented or object based programming language ?
A. Object Based programming language **does not support** all features of OOPs like Polymorphism and Inheritance. Java is an object oriented programming language. Some of object based programming languages are JavaScript, Visual Basics.

Q. Explain Encapsulation with example ?
A. Encapsulation is a mechanism where the data and the code that act on the data will **bind** together.
Example : Class is an example of encapsulation. As class is the collection of data members and member functions, hence class binds them together.

Q. Explain Abstraction with example ?
A. Abstraction refers to the act of **hiding** the background details and providing only necessary details to the user.
Example : Let us the example of a car. While driving car, driver(user) only knows basic operations like gearing system, brakes, but is totally unaware about the internal happenings and processes. This is abstraction.

Q. Explain Polymorphism with example ?
A. Polymorphism represents the **ability** to assume several different forms of a program.
Example : **Overloading** of methods is an example of Polymorphism. Other examples include, abstract classes and interfaces.

Q. How we achieve the concept of polymorphism in Java?
A. Inheritance, Overloading and Overriding are used to achieve Polymorphism in java.

SET 5 – Q21 – Q25

Q. Explain the different forms of Polymorphism.
A. Polymorphism is divided into two forms :

1. Compile time Polymorphism
2. Run time Polymorphism
   Compile time polymorphism is method overloading. Runtime time polymorphism is done using inheritance and interface.

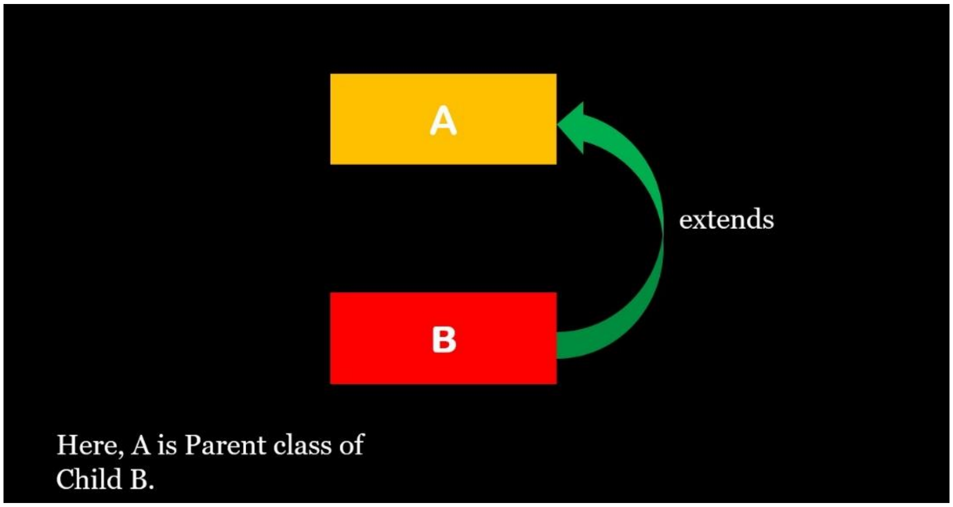
Q. Explain Inheritance with example ?
A. Inheritance refers to the process in which child class inherits the properties of its parent class.
A class that is inherited is called a **superclass**. The class that does the inheriting is called a subclass.
Inheritance is done by using the keyword extends.

Example : Suppose we are having two classes A and B,
class B extends A -> Child class B, is inheriting the property of its parent class.

Q. What are the types of Inheritance shown by Java ?
A. There are four types of Inheritance in Java :

1. Single Inheritance
2. Multilevel Inheritance
3. Hybrid Inheritance
4. Hierarchical Inheritance

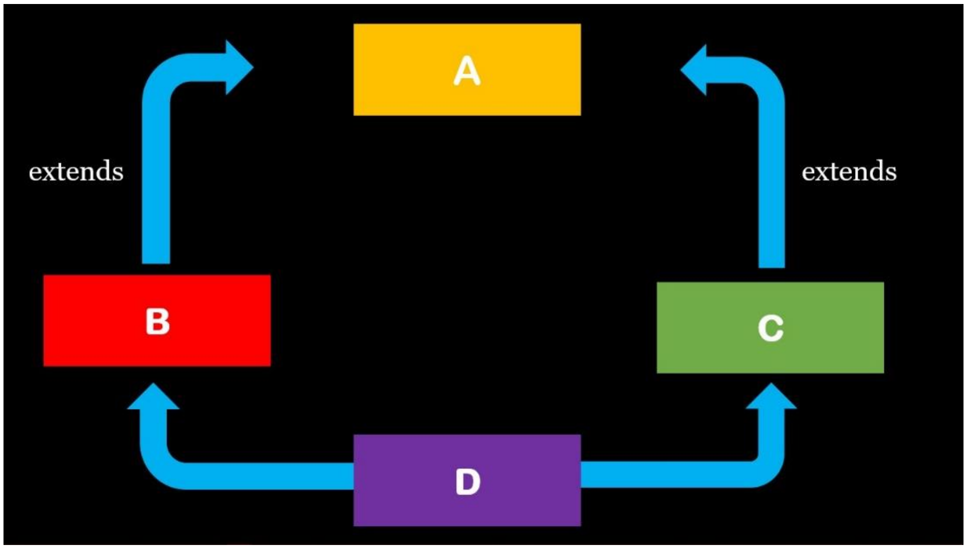
Q. Explain different types of Inheritance ?
A. **Single Inheritance** : It is a type of inheritance, where a single child class inherits the property of its parent class.



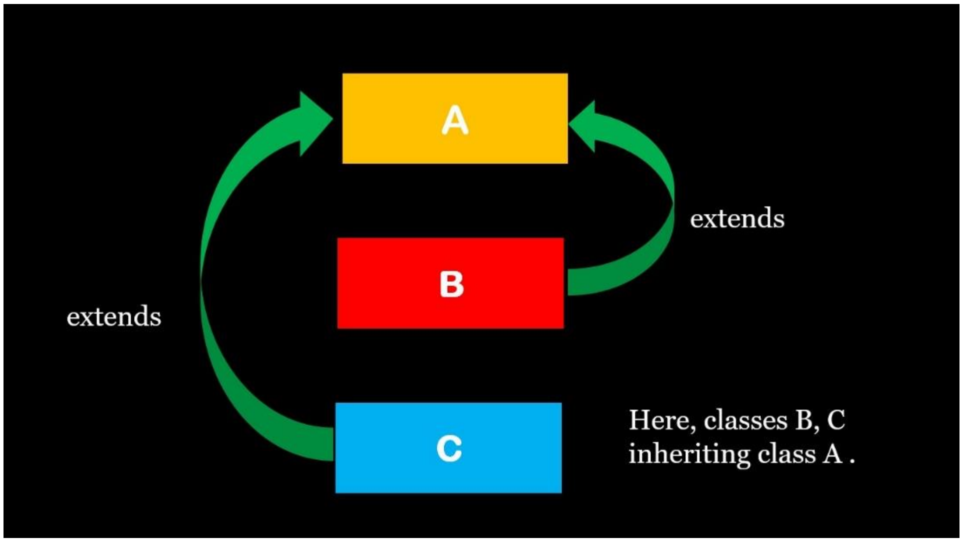
Here, A is Parent class of Child B.

Multilevel Inheritance : A class inherits properties from a class which again has inherits properties.

Hybrid Inheritance : A hybrid inheritance is a combination of more than one types of inheritance.



Hierarchical Interitance : In this type of Inheritance, more than one classes are derived from a single base class or parent class.



Q. Why Java Doesn't Support multiple Inheritance ?
A. This is due to the ambiguity problem.

SET 6 – Q26 – Q30

Q. What happens if super class and sub class having same field name ?
A. Super class fields will be hidden, meanwhile fields of Subclass will be visible. However, you can also access super class fields with super keyword.

Q. Can a children class inherit a constructor of its parent class ?
A. No.

Q. What is Method Overloading ?
A. Method Overloading means to have two or more methods with **same name** in the same class with different arguments. A method can be overloaded in the same class or in a subclass.

Q. What is Method Overriding ?
A. Method overriding occurs when sub class declares a method that has the same type arguments as a method declared by one of its superclass.

Q. Can we override a method marked as final ?
A. **No**, we can not override a method marked with final.

SET 7 – Q31 – Q35

Q. Can we override a method marked as static ?
A. No, we can not override a method marked with static.

Q. Is it possible to override the main method ?
A. No, main is a static method. A static method **can't** be overridden in Java.

Q. Explain the role of implements keyword ?
A. **implements** keyword is used to developed inheritance between interface and class.

Q. Is String a keyword in java?
A. No. **String is not a keyword in java**. String is a final class in java.lang package which is used to represent the set of characters in java.

Q. Can we overload a static method in Java?
A. Yes.

SET 8 – Q36 – Q40

Q. Can we override a static method in Java?
A. No.

Q. Can a class implement more than one interface in Java?
A. Yes.

Q. Can a class extend more than one class in Java?
A. No.

Q. How we can create object without using new operator ?
A. Using factory methods:
NumberFormat obj = **NumberFormat**. getNumberInstance( );
Here, we are creating NumberFormat object using the factory method **getNumberInstance()**.

Q. What are reference variables ?
A. Reference variable usually points to object. In other words, a reference variable is associated with an object. For example :
A ob = new **A()**;
Here, ob is a reference variable for a class A.
Reference variable are like pointers in C.

SET 9 – Q41 – Q45

Q. Explain Mutable and Immutable object in Java ?
A. Mutable objects are the objects which can be modified. For example : **StringBuffer.**
Immutable objects are the objects which can not be modified. For example : **String.**

Q. Where objects are stored in memory ?
A. Objects are stored in **heap** memory.

Q. What is memory leak ?
A. A Memory Leak is a situation when there are objects present in the heap that are no longer used, still, garbage collector is unable to remove them from memory and, thus they are unnecessarily maintained.

Q. Explain the difference between an object and an instance ?
A. An Object **may or may not** have a class definition.
Example – int arr[] where arr is an array.
An Instance should have a class definition.
Example – Solution sol = new Solution(); Here, sol is an instance.

Q. Explain the concept of Singleton Class ?
A. In object-oriented programming, a singleton class is a class that can have only one object at a time.

SET 10 – Q46 – Q50

Q. What is the role of super keyword ?
A. **Super** keyword is used to access the method or member variables from the superclass. If a method overrides one of the methods in its superclass, the method can invoke the overridden method through the use of the super keyword.

Q. What is an Interface ?
A. An **Interface** in the Java programming language is an abstract type that is used to specify a behavior that classes must implement.
Defining an Interface :

```
access interface name {
return-type method-name1();
return-type method-name2();
type final var1 = value;
}
```

An Interface cannot extend anything but another interfaces. Interfaces can not be final.

Q. Can we instantiate an interface ?
A. No, You can't **instantiate** an interface directly, but you can instantiate a class that implements an interface.

Q. What is an abstract class ?
A. Abstract classes are the classes that usually contain **one or more** abstract methods. An abstract method is a method that is declared, but contains no implementation. Whenever abstract keyword is used in front of a class, then it becomes an abstract class.

If even a single method is abstract, the whole class must be declared abstract.

Q. Can we instantiate an abstract class ?
A. No.

SET 11 – Q51 – Q55

Q. Can a class be marked with abstract & final ?
A. No, not possible.

Q. When you declare a method as abstract, can other non abstract methods access it?
A. Yes.

Q. Can there be an abstract class with no abstract methods in it ?
A. Yes.

Q. Explain Constructor ?
A. Constructor is defined as a **special method** that is used for initializing the objects of a class. Constructor has same name as its class. Constructor does not have any return type.

Q. How are this() and super() used with constructors ?
A. Constructors use **this** to refer to another constructor in the same class with different parameters. Constructors use **super** keyword to invoke the superclass's constructor.

SET 12 – Q56 – Q60

Q. What is final modifier?
A. Whenever a variable is marked with final, it means that its value **can not be changed**.

Q. What is static block?
A. Static block is a block of code, which is executed **exactly once** i.e when the class is first loaded into JVM. Before main method the static block executes.
Syntax :

```
static
{
System.out.println(" This is a static block.");
}
```

Q. What are Static variables ?
A. It is a variable which belongs to the class and **initialized only once** at the start of the execution.

Q. A static method can only call other static methods. True or False ?
A. True.

Q. A static method can reference to the current object using keywords super or this. True or False ?
A. False.

SET 13 – Q61 – Q65

Q. What are instance variables ?
A. Instance variables are the variables used within a class but **outside** a method. In this, each instance of class, contains its own copies of these variables.

Q. What if we include a return type to a constructor ?
A. No error, as it will be considered as a method.

Q. Can a method name be same as class name ?
A. Yes.

Q. What is Constructor Overloading ?
A. **Constructor overloading** is the process of defining more than one constructors, each one having different parameters.

Q. Do we have destructors in java ?
A. No. Same job is done through Garbage Collection.

SET 14 – Q66 – Q70

Q. Explain the role of garbage collector ?
A. When no reference to an object exists, that object is needed to be no longer needed, and the memory occupied by the object can be reclaimed. There is no need to explicitly destroying the objects. This is the whole process of garbage collection in java.

Q. What are packages in Java ?
A. **Package** in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. It is achieved through package keyword.

Q. What is S.C.P ?
A. String Constant Pool is the memory space in heap memory specially allocated to store the string objects created using string literals. In String Constant Pool, there will be no two string objects having the same content.

Q. Which is the final class in these three classes – String, StringBuffer and StringBuilder?
A. All three are final.

Q. Which one will you prefer among == and equals() method to compare two string objects?
A. equals() method because it compares two string objects based on their content. If you use == operator, it checks only references of two objects are equal or not.

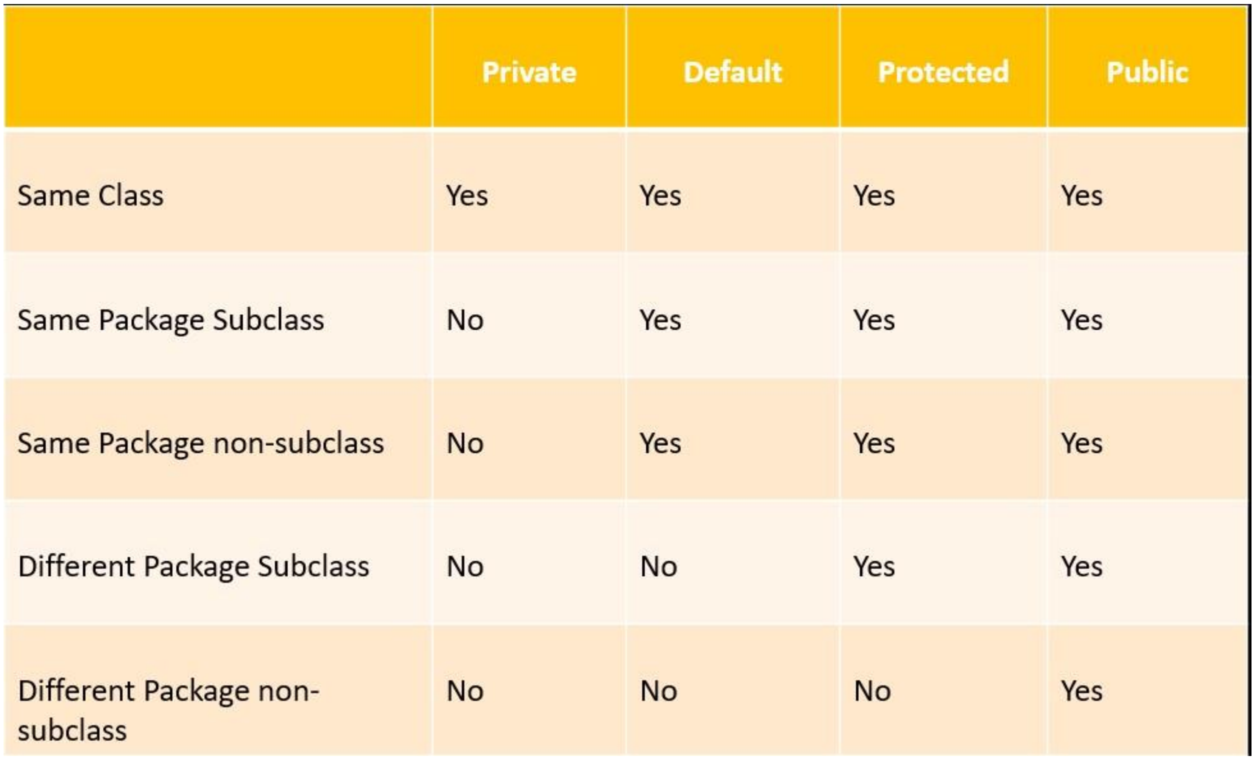
SET 15 – Q71 – Q75

Q. Which class will you recommend among String, StringBuffer and StringBuilder classes if I want mutable and thread safe objects?
A. StringBuffer.

Q. What is the similarity and difference between String and StringBuffer class?
A. The main similarity between **String** and **StringBuffer** class is that both are thread safe. The main difference between them is that String objects are immutable where as StringBuffer objects are mutable.

Q. What is the similarity and difference between StringBuffer and StringBuilder class ?
A. The main similarity between **StringBuffer** and **StringBuilder** class is that both produces mutable string objects. The main difference between them is that StringBuffer class is thread safe where as StringBuilder class is not thread safe.

Q. Explain Class Member Access ?
A.



	Private	Default	Protected	Public
Same Class	Yes	Yes	Yes	Yes
Same Package Subclass	No	Yes	Yes	Yes
Same Package non-subclass	No	Yes	Yes	Yes
Different Package Subclass	No	No	Yes	Yes
Different Package non-subclass	No	No	No	Yes

Q. What is an exception ?
A. An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program’s instructions.

SET 16 – Q76 – Q80

Q. What is an error ?
A. An Error indicates that a non-recoverable condition has occurred that should not be caught.

Q. Which is superclass of Exception ?
A. **Throwable** Class, is the parent class of all exception related classes.

Q. How exceptions can be handled in Java ?
A. Using **try**, **catch**, **throw**, **throws**.

Q. Explain the role of finally block ?
A. It creates a block of code, that will be executed after a try/catch block or before. It will be executed everytime, no matter whether an exception is thrown or not.

Q. What are the types of Exceptions ?
A. **Checked** and **Unchecked**.

SET 17 – Q81 – Q85

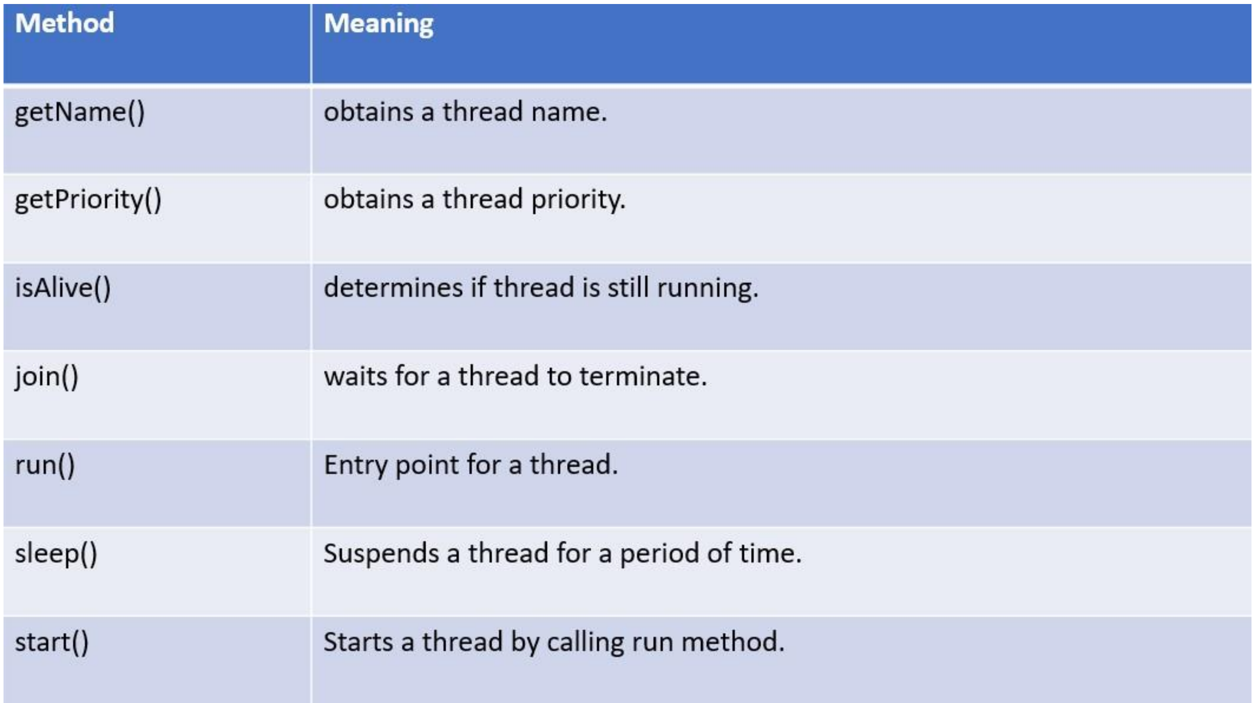
Q. Can we have the try block without catch block ?
A. Yes, we can have the try block without catch block, but finally block should follow the try block.

Q. What is a Thread ?
A. A Thread refers to the smallest unit of processing.

Q. How can you gain the concept of multithreading in Java ?
A. In either of two ways :

1. Extending Thread class
2. Implementing Runnable interface.

Q. Explain some methods that are used in Multithreading ?
A.



Method	Meaning
getName()	obtains a thread name.
getPriority()	obtains a thread priority.
isAlive()	determines if thread is still running.
join()	waits for a thread to terminate.
run()	Entry point for a thread.
sleep()	Suspends a thread for a period of time.
start()	Starts a thread by calling run method.

Q. Explain Thread Synchronization ?
A. When two or more threads needs access to a shared resource, they need some way to ensure that the resource will be used by only one thread at a time. This is Synchronization.
It is achieved through synchronized keyword in Java.

SET 18 – Q86 – Q90

Q. What is the difference between yield() and sleep()?
A. yield() allows the current thread to release its lock from the object and scheduler gives the lock of the object to the other thread with same priority. sleep() allows the thread to go to sleep state for x milliseconds. When a thread goes into sleep state it doesn’t release the lock.

Q. What is the difference between wait() and sleep()?
A. wait() is a method of Object class. sleep() is a method of Object class. sleep() allows the thread to go to sleep state for x milliseconds. When a thread goes into sleep state it doesn’t release the lock. wait() allows thread to release the lock and goes to suspended state. The thread is only active when a notify() or notifyAll() method is called for the same object.

Q. What is difference between notify() and notifyAll()?
A. **notify( )** wakes up the first thread that called wait( ) on the same object.
notifyAll( ) wakes up all the threads that called wait( ) on the same object. The highest priority thread will run first.

Q. What is volatile keyword ?
A. **Volatile** keyword is used to modify the value of a variable by different threads. It is also used to make classes thread safe.

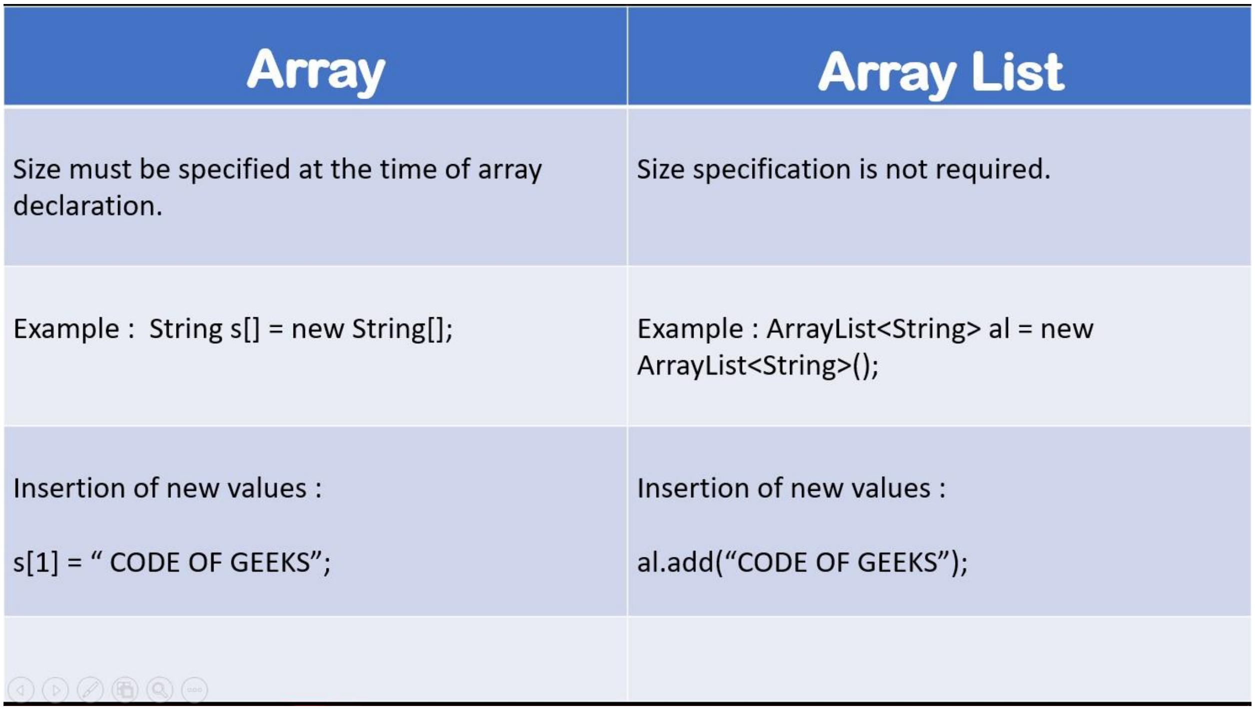
Q. What are the daemon threads?
A. **Daemon threads** are special threads that run in the background, however, these not used to run the application code generally.

SET 19 – Q91 – Q95

Q. How many locks does an object have?
A. Each object has only one lock.

Q. What are Collections in Java ?
A. The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects. Java Collections can achieve all the operations like searching, sorting, insertion, manipulation, and deletion.

Q. Differentiate b/w Array & ArrayList in Java ?
A.



Array	Array List
Size must be specified at the time of array declaration.	Size specification is not required.
Example : String s[] = new String[];	Example : ArrayList<String> al = new ArrayList<String>();
Insertion of new values : s[1] = “ CODE OF GEEKS”;	Insertion of new values : al.add(“CODE OF GEEKS”);

Q. Explain checked & unchecked exceptions ?
A. Checked Exception : Exceptions are checked by the compiler at the time of compilation, are checked exceptions.
Example – ClassNotFound Exception

Unchecked Exception : These exceptions are not checked during the compile time by the compiler.
Example – ArrayIndexOutOfBounds Exception

Q. What is the role of finalize() ?
A. finalize() method is a protected and non-static method of java.lang.Object class. This method is used to perform some final operations or clean up operations on an object before it is removed from the memory.

So, these are most frequently asked interview questions on JAVA.
We, at CODE OF GEEKS, wish you all the best for your upcoming future.
For any assistance, feel free to mail us at support@codeofgeeks.com