## 

## **Q.1)What is Java?**

## Java is a platform-independent high-level programming language. It is platform-independent because its byte codes can run on any system regardless of its operating system.

## **Q.2)Why is Java called platform independent?**

### Java is platform independent that means we can execute our code in any operating system either it is mac, Window or Linux. Java is Platform independent because when we write code, then its compiler converts it into bytecode and this bytecode can be executed on any platform (JDK should be installed in that OS).

### **Q.3)What are the principles of object oriented programming, such as OOPs concepts in Java**

Object-oriented programming (OOP) principles in Java include encapsulation, inheritance, polymorphism, and abstraction.

* Encapsulation: Bundling data and methods together into a single unit (class) to hide implementation details.
* Inheritance: Creating new classes (subclasses) from existing classes (superclasses) to inherit properties and behaviors.
* Polymorphism: The ability of an object to take on many forms, allowing objects of different classes to be treated as objects of a common superclass.
* Abstraction: Simplifying complex systems by breaking them down into smaller, more manageable units.

### **Q.4)Name and explain the types of constructors in Java**

### The two types of constructors in Java are the Default Constructor and the Parameterized Constructor.

### Default Constructor

### Does not take any inputs

### Main purpose is to initialize the instance variables with the default values

### Widely used for object creation

### Parameterized Constructor

### Capable of initializing the instance variables with the provided values.

### These constructors take the arguments.

### **Q.5) What is JDK?**

### JDK stands for Java development kit.

### It can compile, document, and package Java programs.

### It contains both JRE and development tools.

### **Q.6) What is JVM?**

### JVM stands for Java virtual machine.

### It is an abstract machine that provides a run-time environment that allows programmers to execute Java bytecode.

### JVM follows specification, implementation, and runtime instance notations.

### **Q.8) What is JRE?**

### JRE stands for Java runtime environment.

### JRE refers to a runtime environment that allows programmers to execute Java bytecode.

### JRE is a physical implementation of the JVM.

### **Q.9)In Java, what are the differences between heap and stack memory?**

### *Memory*

### Stack memory is used only by one thread of execution.

### All the parts of the application use heap memory.

### *Access*

### Other threads can’t access stack memory.

### Objects stored in the heap are globally accessible.

### *Memory Management*

### Stack follows the LIFO manner to free memory.

### Memory management for heap stems from the generation associated with each object.

### *Lifetime*

### Stack exists until the end of the execution of the thread.

### Heap memory lives from the start till the end of application execution.

### *Usage*

### Stack memory only contains local primitive and reference variables to objects in heap space.

### Whenever you create an object, it is always stored away in the heap space.

### **Q.10)What is a JIT compiler?**

### A JIT compiler runs after the program is executed and compiles the code into a faster form, hosting the CPU’s native instructing set.

### **Q.11) How does a JIT compiler differ from a standard compiler?**

### JIT can access dynamic runtime information, and a standard compiler does not. Therefore, JIT can better optimize frequently used inlining functions.

### 

### **Q.12 Describe the use of the “Static” keyword in Java?**

### Ans. The static keyword in Java is a non-access modifier useful for memory management. We can directly access the static keyword using the class name without creating an instance of the class.

### **Q.13 What is the significance of the super keyword in Java?**

### Ans. The super keyword refers to objects that belong to a superclass (parent). It is used to access the superclass constructor and call superclass methods. The super keyword is most commonly used to distinguish between superclasses and subclasses that have methods with the same name.

### 

### **Q.14.What is Polymorphism?**

Polymorphism is the ability of an object to take many forms. It allows performing the same action in many different ways. Polymorphism in Java has two types: Compile time polymorphism (static binding) and Runtime polymorphism (dynamic binding).

### **Q.15.What is runtime polymorphism or dynamic method dispatch?**

Runtime polymorphism or Dynamic Method Dispatch is a process in which a call to an overridden method is resolved at runtime rather than compile-time. It is called as through the reference variable of a superclass.

### **Q16. What is abstraction in Java?**

Data Abstraction is the property by virtue of which only the essential details are displayed to the user. We use abstract class and interface to achieve abstraction.

### **17. What do you mean by an interface in Java?**

An interface in the Java programming language is an abstract type that is used to specify a behavior that classes must implement. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Method bodies exist only for default methods and static methods.

### 5. What is the difference between abstract classes and interfaces?

An abstract class allows creating functionality that subclasses can implement or override. An interface only allows defining functionality, not implementing it and whereas a class can extend only one abstract class, it can take advantage of multiple interfaces.

### **18 What is inheritance in Java?**

Java Inheritance is a mechanism in which one class acquires the property of another class. When an “Is-A” relationship exists between two classes, we use Inheritance. The parent class is called a super class and the inherited class is called a subclass

### **Q.19) What are the different types of inheritance in Java?**

* Single Inheritance
* Multiple Inheritance
* Multilevel Inheritance
* Hierarchical Inheritance
* Hybrid Inheritance

### **Q.20)What is method overloading and method overriding?**

| Method Overloading | Method Overriding |
| --- | --- |
| It is used to increase the readability of the program | Provides a specific implementation of the method already in the parent class |
| It is performed within the same class | It involves multiple classes |

### **Q.21). Can you override a private or static method in Java?**

No, we cannot override private or static methods methods declared as private can never be overridden, they are in-fact bounded during compile time. Private methods in Java are not visible to any other class which limits their scope to the class in which they are declared.

### **Q.22) What is multiple inheritance? Is it supported by Java?**

Multiple Inheritance is a feature of object oriented concept, where a class can inherit properties of more than one parent class. The problem occurs when there exist methods with same signature in both the super classes and subclass.

1. What does an interface in Java refer to?

* An interface as it relates to Java is a blueprint of a class or a collection of abstract methods and static constants.
* Each method is public and abstract, but it does not contain any constructor.

### **Q.23 What is encapsulation in Java?**

Encapsulation is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. It is the process of hiding information details and protecting data and behavior of the object.

### **Q.24) What is composition in Java?**

Composition is a restricted form of Aggregation in which two entities are highly dependent on each other. It represents part-of relationship. In composition, both entities are dependent on each other. When there is a composition between two entities, the composed object cannot exist without the other entity.

## **Q.25) What do you understand by Exception Handling?**

Exception handling is a process of handling exceptions occurs during the execution of a program. Due to the occurrence of exception, execution of programs get halted, so it is very important to handle these exceptions so that program can be executed smoothly. We can handle the exceptions by using five keywords: try, catch, throw, throws, and finally.

## **Q.26) What is checked and unchecked exception?**

* Checked exception: If the exception occurs or checked at compile time during the execution of a program, it is called as the checked exception. We should handle these exceptions using try-catch block or using throws keyword.  
  E.g., if someone tries to read a file which is not present then it will throw a checked exception at compile time FileNotFoundException
* Unchecked exceptions: If the exception is not checked at compile time and occurred at runtime then this type of exception is called an unchecked exception. This type of exceptions occur due to an error in the logic of the code. If we do not handle this type of exception then also compiler will not give a compilation error.  
  E.g. ArithmeticException

## **Q.27) What are the reasons behind the occurrence of an exception?**

Following are the reasons behind the occurrence of an exception:

* Accessing a file, which does not exist
* Dividing a variable by zero
* Inserting an element in the array outside the range
* If throw statement occurs
* Abnormal execution condition captured by JVM

### **Q.28)What is a copy constructor in Java?**

Copy constructor is a special type of constructor that creates an object using another object of the same Java class. It returns a duplicate copy of an existing object of the class. We can assign a value to the final field but the same cannot be done while using the clone() method.

**Q.29)What is the difference between ArrayList and LinkedList?**

ArrayList is a resizable array implementation, while LinkedList is a doubly linked list implementation.

* ArrayList uses an array to store elements, allowing fast random access but slower insertions and deletions.
* LinkedList uses nodes with references to previous and next elements, allowing fast insertions and deletions but slower random access.
* ArrayList is more suitable for scenarios where random access is frequent, while LinkedList is better for frequent insertions and deletions.
* Example: ArrayList can be used to store a list of students' names, while LinkedList can be used to implement a queue.

## **Q.30)Differentiate between class and object.**

The class and object both are the features of OOPs concepts. The basic differences between both features are given below:

* The Class is a logical entity whereas Object is a physical quantity.
* Class does not occupy memory at the time of creation whereas Object occupied space in memory when it is created.
* For declaring a class, we use a 'class' keyword followed by a class name, whereas we can create the object using the 'new' keyword in Java.
* A Class is like a factory which generates object and object are the instances of the class.

## **Q.31)What is encapsulation in Java?**

Encapsulation is a process of enclosing the data and code together to form a single unit. The best example to understand the encapsulation is a capsule which contains the medicine in it.

* If we declare all the data members of the class as private, then it is called a fully encapsulated class in Java, and then we can use getter and setter method to access it.
* One of the examples of the fully encapsulated class is Java Bean class.
* Encapsulation keeps its data hide from other class hence it is also called as data-hiding.

## **Q.32)What do you understand by runtime polymorphism?**

Polymorphism is a method of performing "a single task in different ways." Polymorphism is of two types

1. Runtime Polymorphism
2. Compile-time polymorphism

Runtime Polymorphism- We can achieve runtime Polymorphism by method overriding in Java. And method overriding is a process of overriding a method in the subclass which is having the same signature as that of in superclass.

## **Q.33)How can you differentiate between method overloading and method overriding?**

| No. | Method overloading | Method overriding |
| --- | --- | --- |
| 1. | The process of calling two methods having the same name with different parameters is called method overloading (in the same class) | The process of calling two methods, one in the subclass and other in the superclass, having the same signature is called as method overriding. |
| 2. | It can be accessed within a class. | Method overriding requires two classes to be accessed which having IS-A relationship. |
| 3. | Return type may be changed or may remain same with different parameters | Return type should be the same for both methods. |
| 4. | Method overloading is a concept of compile-time polymorphism. | Method overriding is a concept of method overriding. |
| 5. | e.g. class A{  void m1()  {// codes.......}  Void m1 (int a) {//code.........} | e.g. class A {  void m1(){  // code............}  }  Class B extends A{  Void m1(){  // code...........} |

## **Q.34) What are the keyword "super" and "this" in Java?**

super keyword: "super" is a keyword in Java which is used to give reference to the object of parent class. "super" keyword cannot be used as an identifier as it is reserved keyword in Java.

this Keyword: "this" keyword in Java is used to refer to the object of the current class. The 'this' keyword cannot be used as an identifier as it is a reserved keyword in Java.

## **Q.35)What is an interface in Java? Can we implement multiple interfaces in one class?**

Interface in Java is a way to achieve the abstraction. The Interface is like a class but not exactly as Interface also can have methods and variable as the class does but Interface only contain method signature does not have the body.

* The Interface cannot be instantiated in Java.
* The Interface contains methods which are public and abstract (by default).
* A class can implement an interface.
* For declaring an interface, we use the keyword interface.

Syntax:

* interface Interface\_Name{
* //Methods
* }

We can implement multiple interfaces in one class and parent interfaces can be declared using a comma(,) operator.

Syntax:

* public class A implements C, D {
* Code
* }

## **Q.36) Explain inheritance in Java? How can it be achieved?**

* Inheritance in Java is a process by which one class can have all properties of other class. That means one class inherits all the behaviour of the other class.
* Inheritance increases the code reusability.
* Inheritance is an important feature of OOP concept.
* Inheritance is also a representation of the IS-A relationship

There are two terms used in inheritance:

1. Child class (Subclass): Class which inherits other class, called a Child class or derived class.
2. Parent class (Superclass): A class which got inherited by another class is termed as parent class or Base class.

The Syntax of java inheritance:

* Class A extends B // Here A represents subclass and B represent Superclass
* {
* // Code
* }

## **Q.37) Can we use multiple inheritance in Java? Explain with reason?**

No, we cannot use multiple inheritance in java as it creates ambiguity and diamond problem in the program. To overcome this problem, we can use interface in Java.

Let suppose class A inherits the two parent class B and C in which a method with the same name is present in both the classes and hence when we try to override that method it will create confusion for the compiler and will give the compilation error. Therefore, Java doesn?t support multiple inheritance.

## **Q.38) What can we do if we want to access a private member of a class?**

We can access private members of the class by using public getters and setters from outside the class in Java.

## **Q.39)What is the significance of "static" keyword?**

* Static keyword in Java is a non-access modifier which can be used with the block, variable, methods, and nested classes.
* Static Keywords are the part of the class, and it does not belong to the instance of the class.
* We use static keyword in java with variables, block, and method for achieving memory management.
* Java static property can be shared by all the objects.
* For accessing the static members, we don't need to create the instance of the class.

## **Q.40)What is "Collection Framework" in Java?**

Collection Framework in Java is an architecture for storing the classes, and interfaces and manipulating the data in the form of objects. There are two main interfaces in Collection Framework that are:

* Java.util.Collection
* Java.util.Map

**Q.41)What is an inner class?**

## An inner class is a class that is nested within another class. An Inner class has access rights for the class that is nesting it, and it can access all variables and methods defined in the outer class.

## **Q.42)What is a subclass?**

## A subclass is a class that inherits from another class called the superclass. Subclass can access all public and protected methods and fields of its superclass.

## **Q.43)What is a package in Java?**

## In Java, packages are the collection of related classes and interfaces which bundle together.

## **Q.44) How can developers use packages in Java?**

## Packages in Java allow developers to modularize the code and optimize its reuse easily. In addition, developers can use other classes to import and reuse the code within the packages.

## **Q.45)What are the advantages of packages in Java?**

## Packages help developers avoid name clashes.

## Packages provide easier access control.

## Packages can also contain hidden classes that are not visible to the outer classes and are only used within the package.

## Packages create a standardized hierarchical structure, making it easier to locate related classes.

## **Q.46)What are Java data types, and how are they grouped?**

## In Java, a variable must be a specified data type such as an integer, floating-point number, character Boolean, or string. The two groups of data types are:

## Primitive data types, which include byte, short, int, long, float, double, boolean, and char

## Non-primitive data types, which include string, arrays, and classes

## **Q.47)What do the terms autoboxing and unboxing mean in Java?**

## *Autoboxing* represents the Java compiler that automatically transforms primitive data types into object equivalents or wrapper types to ease compilation.

## *Unboxing* is the automatic transformation of wrapper types into their primitive equivalent.

## **Q.48) What are wrapper classes in Java?**

## Every primitive data type has a class dedicated to it, known as wrapper classes.

## We call them wrapper classes because they “wrap” the primitive data type into an object of that class.

## Wrapper classes convert the Java primitives into reference types (objects).

## **Q.49)Can you override a private method or static method in Java?**

## You cannot override a private or static method in Java. You cannot override a private method in subclass because it’s not accessible there.

## **Q.50)What is method hiding?**

## Method hiding is an alternative to overriding a private or static method, which occurs when you hide the superclass method. You create a similar method with the same return type and same method arguments in child class. For example, you can create another private method with the same name in the child class.

## **Q.51)What is the difference between equals() and == in Java?**

## Equals() method

## Is defined in object class in Java.

## Used for checking the equality of two objects defined by business logic.

## “==” (equality operator)

## A binary operator provided by Java programming language and used to compare primitives and objects.

## public boolean equals (object o) is the method provided by the Object class.

## Default uses == operator to compare two objects. For example, you can override a method like string class. equals() method is used to compare the values of two objects.

## **Q.52)Can you write multiple catch blocks under a single try block?**

## Yes, you can have multiple catch blocks under a single try block. Your approach should be from specific to general, as shown in the following example:

## **Q.53)What is a local variable?**

## Local variables are defined in the method and scope of the variables that exist inside the method itself.

## **Q.54) What is an instance variable?**

## An instance variable is defined inside the class and outside the method. The scope of the variables exists throughout the class.

## **Q.55) How do you use final keywords and final variables in Java?**

## When Java programmers use final keywords with a variable of primitive data types, they cannot change the variable’s value.

## When you use final with non-primitive variables, you cannot change the members of the referred object.

## 

## **Q.56)What is an infinite loop?**

## An infinite loop runs without any condition and runs infinitely. You can break an infinite loop by defining any breaking logic in the body of the statement blocks.

## How do you declare an infinite loop?

## for (;;)

## {

## // Statements to execute

## // Add any loop breaking logic

## }

## **Q.57)What is the difference between the continue and break statement?**

## Break and continue are two important keywords used in loops. When using a *break* keyword in a loop, the loop breaks instantly. The current iteration breaks when using the *continue* keyword, and the loop continues with the next iteration.

## **Q.58)What is the entry point in Java, and how is it written?**

## main() in Java is the entry point for any Java program.

## main() is always written as public static void main string args.

## **Q.59)In Java, what are public static void main string args?**

## Public static void main string args, also known as public static void main(String[] args), means:

## *Public* is an access modifier used to specify who can access this method. Also, this method is accessible by any class.

## *Static* is a keyword in java that identifies when it is class-based. main() is made static in Java to access it without creating the instance of a class. If main is not made static, the compiler will throw an error as main() is called by the JVM before creating any objects. It can only invoke static methods directly via the class.

## *Void* is the return type of the method that defines the method. That method does not return a value.

## *Main* is the name of the method searched by JVM as a starting point for an application (with a particular signature only). It is also the method where the main execution occurs.

## *String args[]* is the parameter that passes to the main method.

## In Java, what’s the purpose of static methods and static variables?

## Developers use a static keyword to make a method or variable shared for all objects when there is a requirement to share a method or a variable between multiple objects of a class. This is used instead of creating separate copies for each object.

## **Q.60)What does this() represent, and how is it used in Java?**

## this() represents the current instance of a class

## Used to:

## Call the default constructor of the same class

## Access methods of the current class

## Point to the current class instance

## **Q.61)What does super()represent, and how is it used in Java?**

## super() represents the current instance of a parent/base class

## Used to:

## Call the default constructor of the parent/base class

## Access methods of the base class

## Point to the superclass instance

## **Q.62) What is a Java switch statement, and how can it be used?**

## As a standard programming logic, it can simply be achieved by using if…else conditions.

## In programs involving more complicated cases, complex scenarios require calling several methods for which switch solves this problem.

## Switch avoids several nested if…else statements.

## In Java scenarios that yield a high number of iterations, the switch is typically faster than using if….else statements.

## **Q.63)What is the default of the switch case?**

## In a switch statement, the default case executes when no other switch condition matches. Because the default case is optional, you can only declare it after coding all other switch cases.

## This comprehensive list of Java interview questions should have you better prepared for an interview, and not only that but a successful one. Make sure to write down your answers as you prepare, so you can better remember them. Ensure you practice — even in front of a mirror. Knowing how you sound and appear will only boost your confidence and make you better able to land that dream Java job you’ve been wanting.

## 

## **Q.64)What is List interface in collections?**

List interface is an interface in Java Collection Framework. List interface extends the Collection interface.

* It is an ordered collection of objects.
* It contains duplicate elements.
* It also allows random access of elements.

Syntax:

* public interface List<E> extends Collection<E>

**Q.65) What do you understand by object cloning?**

Object cloning is a mechanism of creating the same copy of an object. For object cloning, we can use clone() method of the Object class. The class must implement the java.lang.Clonable interface, whose clone we want to create otherwise it will throw an exception.

Syntax of clone() method:

* protected Object clone() throws CloneNotSupportedException

## **Q.66) Can we insert duplicate values in Set?**

We cannot insert duplicate elements in Set. If we add a duplicate element, then the output will only show the unique elements.

## **Q.67) What is the difference between Collections, and Collection in Java?**

Collection and collections both are the part of Java Collection Framework, but the primary differences between both are given below:

* A Collection is an interface in java and Collections is a class of collection framework.
* The Collection interface provides the methods that can be used for data structure whereas Collections class provides the static methods which can be used for various operation on a collection.

## **Q.68) What is "Diamond problem" in Java? How can it be removed?**

The Diamond problem occurs in multiple inheritance, but Java does not allow multiple inheritance. In case of Java, it can occur with interfaces. When we implement two interfaces which are having methods with the same signature then it creates ambiguity for the compiler, and it gives compile time error. Its structure looks like diamond so it is called as Diamond problem.

E.g. Let's take an example which will show the diamond problem.

* interface InterfaceA {
* default public void m1() { System.out.println("This is interface A!"); }
* }
* interface InterfaceB {
* default public void m1(){ System.out.println("This is interface B!"); } //same signature as interface InterfaceA?
* }
* public class Simple implements InterfaceA, InterfaceB {
* public static void main(String args[]) {
* Simple s1= new Simple();
* s1.m1(); // It will give error..
* }}

Error: Simple.java:10: error: class Simple inherits unrelated defaults for m1() from types InterfaceA and InterfaceB

## **Q.69)What is an abstract class in Java?**

* An Abstract class is used to achieve abstraction in Java. If we use the keyword "abstract" with the class name, then it is called as an abstract class.
* An Abstract class can have only methods without body or can have methods with some implementation.
* The Abstract class cannot be instantiated
* It's not necessary that an abstract class should have an abstract method.

Syntax:

* abstract class Student{
* }

## Q**.70)What is deadlock condition in multithreading?**

A Deadlock condition occurs in the case of multithreading. It is a condition when one thread is waiting for an object lock, which is already acquired by another thread and the second thread is waiting for lock object which is taken by the first thread, so this is called deadlock condition in Java.

## **Q.71) Differentiate between Collection and array.**

The Collection and Array both are responsible for storing the data, but the fundamental differences between both are given below:

* Arrays are always of fixed size, we cannot change its size at runtime, but In Collection, size can be changed dynamically.
* Arrays can only store homogeneous or similar type objects, but in Collection, both homogeneous and heterogeneous objects can be stored.
* Arrays cannot provide the "ready-made" methods for user requirements as sorting, searching, etc. but Collection includes readymade methods to use.
* Arrays are good in performance as compare to Collection but Array take more space in memory in comparison to Collection.

### **Q.72)What is the difference between classes and interface?**

The differences between classes and interfaces are listed below:

* A class can be instantiated by creating its object, whereas interfaces cannot be instantiated as all the methods in the interface are abstract and do not perform any action, so there is no use of instantiating an interface.
* A class is declared using class keyword whereas an interface is declared using interface keyword.
* The members of the class can have access specifier such as public, protected, and private but members of the interface can not have the access specifier, all the members of the interface is declared as public because the interface is used to derive another class. There will be no use to access specifies inside the members of an interface.
* The methods inside the class are defined to perform some actions on the fields declared in the class whereas interface lacks in asserting in areas, the ways in an interface are purely abstract.
* A class can implement any number of the interface but can only extend one superclass. Whereas interface can reach any number of interfaces but cannot perform any interface.
* A class can have a constructor defined inside the class to declare the fields inside the class, whereas interface doesn't have any constructor defined because there are no fields to be initialized.

### **Q.73)** What do you mean by aggregation?

### **The term aggregation refers to the relationship between two classes best described as a “whole/part” and “has-a” relationship. This kind is the most specialized version of an association relationship. It contains the reference to another class and is said to have ownership of that class.**

### 

**Q.74). Why is Java not a pure object oriented language?**

Java supports primitive data types - byte, boolean, char, short, int, float, long, and double and hence it is not a pure [object oriented language](https://www.interviewbit.com/oops-interview-questions/).

### **Q.75)What is a ClassLoader?**

### A classloader in Java is a subsystem of Java Virtual Machine, dedicated to loading class files when a program is executed; ClassLoader is the first to load the executable file.

### 

### **Q.76)What is a Marker Interface?**

An empty [interface in Java](https://www.simplilearn.com/tutorials/java-tutorial/java-interface) is referred to as a Marker interface. Serializable and Cloneable are some famous examples of Marker Interface.

### **Q.77) What is Object Cloning?**

An ability to recreate an object entirely similar to an existing object is known as Object Cloning in Java. Java provides a clone() method to clone a current object offering the same functionality as the original object.

### **Q.78) Can Java be said to be the complete object-oriented programming language**

No, Java cannot be treated as a complete object-oriented programming language.

### **Q.79)What is a singleton class in Java? And How to implement a singleton class?**

A class that can possess only one object at a time is called a singleton class. To implement a singleton class given steps are to be followed:

1. Make sure that the class has only one object
2. Give global access to that object

### **Q.80)Explain Java String Pool.**

A collection of strings in Java's Heap memory is referred to as Java String Pool. In case you try to create a new string object, JVM first checks for the presence of the object in the pool. If available, the same object reference is shared with the variable, else a new object is created

### **Q.81) What is the final keyword in Java?**

The term final is a predefined word in Java that is used while declaring values to variables. When a value is declared using the final keyword, then the variable's value remains constant throughout the program's execution.

### **Q.82)What happens when the main() isn't declared as static?**

When the main method is not declared as static, then the program may be compiled correctly but ends up with a severe ambiguity and throws a run time error that reads "NoSuchMethodError."

### **Q.83) Which among String or String Buffer should be preferred when there are a lot of updates required to be done in the data?**

Because StringBuilder is quicker than StringBuffer, it is advised to utilize it wherever possible. However, StringBuffer objects are the best choice if thread safety is required.

### **Q.84) What happens if the static modifier is not included in the main method signature in Java?**

The main function is called by the JVM even before the objects are created, thus even if the code correctly compiles, there will still be an error at runtime.

### **Q.85)Can we make the main() thread a daemon thread?**

This technique designates whether the active thread is a user thread or a daemon thread. For instance, tU.setDaemon(true) would convert a user thread named tU into a daemon thread. On the other side, executing tD.setDaemon(false) would convert a Daemon thread, tD, into a user thread.

### **Q.86) What happens if there are multiple main methods inside one class in Java?**

There is no limit to the number of major approaches you can use. Overloading is the ability to have main methods with different signatures than main (String []), and the JVM will disregard those main methods.

### **Q.87) How does an exception propagate in the code?**

In the event that an exception is not caught, it is initially thrown from the top of the stack and then moves down the call stack to the preceding method. The runtime system looks for a way to handle an exception that a method throws. The ordered list of methods that were called to get to the method where the error occurred is the collection of potential "somethings" that can be used to manage the exception. The call stack is the list of methods, and exception propagation is the search technique.

### **Q.88) Is it mandatory for a catch block to be followed after a try block?**

Each attempt block does not necessarily have to be followed by a catch block. Either a catch block or a final block ought to come after it. Additionally, any exceptions that are expected to be thrown should be mentioned in the method's throws clause.

### **Q.89) Can you call a constructor of a class inside another constructor?**

Yes, a class may include any number of constructors, and each function Object () {[native code] } may call the others using the this() function Object() { [native code] } call function [please do not mix the this() function Object() { [native code] } call function with this keyword]. The constructor's first line should be either this () or this(args). Overloading of constructors is what this is called.

### **Q.90)Why does the java array index start with 0?**

The distance from the array's beginning is just an offset. There is no distance because the first element is at the beginning of the array. Consequently, the offset is 0.

### **Q.91) Why is the remove method faster in the linked list than in an array?**

Because there is no background scaling of an array, insertion, addition, and removal operations are quicker with a LinkedList. Only references in adjacent items need to update when a new item is added in the middle of the list.

### **Q.92)How is the creation of a String using new() different from that of a literal?**

The new () operator always produces a new object in heap memory when creating a String object. The String pool may return an existing object if we build an object using the String literal syntax, such as "Baeldung," on the other hand.

### **Q.93) How is the ‘new' operator different from the ‘newInstance()' operator in java?**

Both the new operator and the newInstance() method are used to create objects in Java. If we already know the kind of object to create, we can use the new operator; however, if the type of object to create is supplied to us at runtime, we must use the newInstance() function.

### **Q.94) Is exceeding the memory limit possible in a program despite having a garbage collector?**

Yes, even with a garbage collector in place, the programme could still run out of memory. Garbage collection aids in identifying and removing programme objects that are no longer needed in order to release the resources they use. When an object in a programme cannot be reached, trash collection is executed with respect to that object. If there is not enough memory available to create new objects, a garbage collector is used to free up memory for things that have been removed from the scope. When the amount of memory released is insufficient for the creation of new objects, the program's memory limit is exceeded.

### **Q.95) Why is synchronization necessary? Explain with the help of a relevant example.**

Multiple threads trying to access the same resources in a multi-threaded software may frequently result in unexpected and incorrect outcomes. Therefore, it must be ensured through some form of synchronization that only one thread can access the resource at any given time. Java offers a method for setting up threads and synchronizing their operations with the aid of synchronized blocks. The synchronized keyword in Java is used to identify synchronized blocks. In Java, a synchronized block is one that is tied to an object. Only one thread can be running at a time inside synchronized blocks since they are all synchronized on the same object. Until the thread inside the synchronized block exits the block, all other threads trying to enter the block are blocked.

### **Q.96)Can you explain the Java thread lifecycle?**

A thread can be in any of the following states in Java. These are the states:

* New: A new thread is always in the new state when it is first formed. The function hasn't been run yet, thus it hasn't started to execute for a thread in the new state.
* Active: A thread switches from the new state to the active state when it calls the start() method. The runnable state and the running state are both contained within the active state.
* Blocked or Waiting: A thread is either in the blocked state or the waiting state when it is inactive for a while (but not indefinitely).
* Timed waiting: When we use the sleep () method on a particular thread, we are actually engaging in timed waiting. The thread enters the timed wait state using the sleep () function. The thread awakens when the allotted time has passed and resumes execution where it left off.
* Termination: A thread that has been terminated means it is no longer active in the system. In other words, the thread is inactive and cannot be revived (made active again after being killed).

### **Q.97)Define System.out.println().**

System.out.println() in Java outputs the argument that was supplied to it. On the monitor, the println() method displays the findings. An objectname is typically used to call a method.

### **Q.98)Will the final block be executed if the code System.exit(0) is written at the end of the try block?**

The system is established as the last line to be run, after which nothing will happen, therefore both the catch and finally blocks are essentially ignored.

### **Q.99)What are the possible ways of making objects eligible for garbage collection (GC) in Java?**

If a reference variable for an object is removed from the programme while it is running, the object may be trash collected. They are also referred to as inaccessible objects occasionally. The new operator returns a reference to an object after dynamically allocating memory for it.

### **Q.100)In the below Java Program, how many objects are eligible for garbage collection?**

I don't know about the program, but generally, three objects are eligible for garbage collection.

The first object is created when the program is started and is no longer needed when the program ends.

The second object is created when the user inputs their name and is no longer required when the program ends.

The third object is created when the user inputs their address and is no longer needed when the program ends.

### **Q.101) Is delete, next, main, exit or null keyword in java?**

No, these keywords do not exist in Java. Delete, Next, Exit are the operations performed in the Java program, Main is the predefined method, and Null is the default String type.

### **Q.102)Explain the difference between >> and >>> operators.**

Although they look similar, there is a massive difference between both.

* >> operator does the job of right shifting the sign bits
* >>> operator is used in shifting out the zero-filled bits

### **Q.103)Why are generics used in Java Programming?**

Compile-time type safety is provided by using generics. Compile-time type safety allows users to catch unnecessary invalid types at compile time. Generic methods and classes help programmers specify a single method declaration, a set of related methods, or related types with an available class declaration.

### **Q.104)Explain the Externalizable interface.**

The Externalizable interface helps with control over the process of serialization. An "externalisable" interface incorporates readExternal and writeExternal methods.

**Q.105)What is the Daemon Thread?**

The Daemon thread can be defined as a thread with the least priority. This Daemon thread is designed to run in the background during the Garbage Collection in Java.

The setDaemon() method creates a Daemon thread in Java.

### **Q .106) Explain the term enumeration in Java.**

Enumeration or [enum is an interface in Java](https://www.simplilearn.com/tutorials/java-tutorial/enum-in-java). Enum allows the sequential access of the elements stored in a collection in Java.

### **Q.107) Why is Java is Dynamic?**

Java is designed to adapt to an evolving environment. Java programs include a large amount of runtime information that is used to resolve access to objects in real-time.

### **Q.109)Can you run a code before executing the main method?**

Yes, we can execute any code, even before the main method. We will be using a static block of code when creating the objects at the class's load time. Any statements within this static block of code will get executed at once while loading the class, even before creating objects in the main method.

### **Q.110)How many times is the finalize method called?**

The finalize method is called the Garbage collector. For every object, the Garbage Collector calls the finalize() method just for one time.

### **Q.111)What is JDBC?**

JDBC is an abbreviation for Java Database Connector.

JDBC is an abstraction layer used to establish connectivity between an existing database and a Java application

### **Q.112)What is the difference between System.out, System.err, and System.in?**

System.out and System.err represent the monitor by default and thus can be used to send data or results to the monitor. System.out is used to display normal messages and results. System.eerr is used to display error messages. System.in represents InputStream object which by default represents standard input device, i.e., keyboard.

### **Q.113)Can the main method be overloaded?**

Yes, the main method can be overloaded as many times as we want. Nevertheless, JVM prefers to call the main method with the help of its predefined calling method.

### **Q.114)Do final, finally and finalize keywords have the same function?**

No, final, finally and finalize keywords have different functionalities.

Final is used to restrict classes, variables, or methods, the final keyword.

Finally is used to execute the code written inside the block without handling any exceptions.

Finalize is used to call the function of the implementation of cleaning the garbage collection of an object.

### **Q.115)What are the different ways of thread usage?**

There are two ways to define and implement a thread in Java. They are by implementing the runnable interface and extending the thread class.

### **Q.116)What is the difference between the ‘throw' and ‘throws' keyword in Java?**

The throw keyword is often used to explicitly throw an exception. It can only throw one exception at a time whereas throws can be used to declare multiple exceptions.

### **Q.117)Difference between static methods, static variables, and static classes in Java.**

A variable, method, or class can be made static by using the static keyword. A static class cannot be instantiated. When both objects or instances of a class share the same variables, this is referred to as static variables. Static methods are simply methods that refer to the class in which they are written.

### **Q.118)How would you differentiate between a String, StringBuffer, and a StringBuilder?**

The string class is immutable but the other two are mutable in nature. StringBuffer is synchronous whereas the StringBuilder is asynchronous. String uses string pool as memory storage whereas the other two use heap memory for storage purposes.

### **Q.119)What is a Comparator in Java?**

A comparator is an interface, which is used to sort the objects.

### **Q.120) What makes a HashSet different from a TreeSet?**

In a HashSet, the elements are unsorted and work faster than a Tree set. It is implemented using a hash table.

### **Q.121)What is the ‘IS-A ‘ relationship in OOPs Java?**

‘IS-A’ relationship is related to the Inheritance property of OOPs Java. It is a kind of parent-child relationship that is established between two classes.

### **Q.122)What is a Java Virtual Machine?**

It is an abstract machine that provides a runtime environment for Java programs to run. The JVM interprets the compiled Java code and executes it on the underlying operating System.

### **Q.123) What is the final method?**

A final method in Java is a method that cannot be overridden in a subclass.

### **Q.124) What is the final class?**

A final class in Java is a class that cannot be subclassed.

### **Q.125) What is the final blank variable?**

The final blank variable in Java is a final variable that is not initialized when it is declared.

### **Q.126)How do you reverse a linked list in Java?**

You can reverse a linked list in Java by iterating through the list and reversing the order of the links between nodes

**Q.127)What is the difference between an abstract class and an interface in Java?**

An abstract class is a class that cannot be instantiated and can contain abstract methods, while an interface is a collection of abstract methods and constants that can be implemented by any class.

### **Q.128)What is a lambda expression in Java? Provide an example.**

A lambda expression in Java is a function that can be created without belonging to any class. An example is (x, y) -> x + y, which takes two integer parameters and returns their sum.

### **Q.129)What is the difference between a private and a protected method in Java?**

A private method is only accessible within the same class, while a protected method is accessible within the same class and any subclass.

### **Q.130)What is the difference between a while loop and a do-while loop in Java?**

A while loop executes the loop body if the condition is true, while a do-while loop executes the loop body at least once before checking the condition.

### **Q.131)What are the different types of Thread Priorities in Java? And what is the default priority of a thread assigned by JVM?**

Priorities in threads is a concept where every thread is having a priority which in layman’s language one can say every object is having priority here which is represented by numbers ranging from 1 to 10. There are different types of thread properties in Java mentioned below:

* MIN\_PRIORITY
* MAX\_PRIORITY
* NORM\_PRIORITY

### **Q.132)What are the steps to connect to the database in Java?**

There are certain steps to connect the database and Java Program as mentioned below:

* Import the Packages
* Load the drivers using the forName() method
* Register the drivers using DriverManager
* Establish a connection using the Connection class object
* Create a statement
* Execute the query
* Close the connections

### **Q.133)Differentiate between Iterable and Iterator.**

| **Iterable** | **Iterator** |
| --- | --- |
| Iterable provides a way to iterate over a sequence of elements. | Iterator helps in iterating over a collection of elements sequentially. |
| **iterator()** method returns an Iterator. | **hasNext()** and **next()** methods are required. |
| **remove()** method is optional. | **remove()** method is required in the iterator. |
| Examples are **List, Queue, and Set.** | Examples are **ListIterator, Enumeration, and ArrayIterator.** |

### **Q.134)Differentiate between the Singly Linked List and Doubly Linked List.**

| **Singly Linked List** | **Doubly Linked List** |
| --- | --- |
| Singly Linked List contain only two segments i.e, Data and Link. | Doubly Linked List contains three segments i.e, Data, and two pointers. |
| Traversal in a singly linked list is possible in only a forward direction. | Traversal in a doubly linked list is only possible in both directions forward as well as backward. |
| It uses less memory as every single node has only one pointer. | It requires more memory than a singly linked list as each node has two pointers. |

### **Q.135)Differentiate between HashSet and TreeSet.**

| **HashSet** | **TreeSet** |
| --- | --- |
| HashSet is unordered. | TreeSet is based on natural ordering. |
| HashSet allows null elements. | TreeSet does not allow null elements. |
| HashSet is Implemented by the HashSet class. | TreeSet is Implemented by TreeSet class. |
| HashSet<String> hs = new HashSet<>(); | TreeSet<String> ts = new TreeSet<>(); |

### **Q.136)What is JVM?**

JVM stands for Java Virtual Machine it is a Java interpreter. It is responsible for loading, verifying, and executing the bytecode created in Java.

Although it is platform dependent which means the software of JVM is different for different Operating Systems it plays a vital role in making Java platform Independent

### **Q.138) Explain public static void main(String args[]) in Java.**

Unlike any other programming language like C, C++, etc. In Java, we declared the main function as a public static void main (String args[]). The meanings of the terms are mentioned below:

1. **public**: the public is the access modifier responsible for mentioning who can access the element or the method and what is the limit. It is responsible for making the main function globally available. It is made public so that JVM can invoke it from outside the class as it is not present in the current class.
2. **static**: static is a keyword used so that we can use the element without initiating the class so to avoid the unnecessary allocation of the memory.
3. **void**: void is a keyword and is used to specify that a method doesn’t return anything. As the main function doesn’t return anything we use void.
4. **main**: main represents that the function declared is the main function. It helps JVM to identify that the declared function is the main function.
5. **String args[]**: It stores Java command-line arguments and is an array of type java.lang.String class.

**Q.139). How many ways you can take input from the console?**

There are two methods to take input from the console in Java mentioned below:

1. Using Command line argument
2. Using Buffered Reader Class
3. Using Console Class
4. Using Scanner Class

Q.140)**Difference in the use of print, println, and printf.**

print, println, and printf all are used for printing the elements but print prints all the elements and the cursor remains in the same line. println shifts the cursor to next line. And with printf we can use format identifiers too.

### **Q.141) What are operators?**

Operators are the special types of symbols used for performing some operations over variables and values.

### **Q.142) How many types of operators are available in Java?**

All types of operators in Java are mentioned below:

1. Arithmetic Operators
2. Unary Operators
3. Assignment Operator
4. Relational Operators
5. Logical Operators
6. Ternary Operator
7. Bitwise Operators
8. Shift Operators
9. instance of operator

**Q.143)What is dot operator?**

The Dot operator in Java is used to access the instance variables and methods of class objects. It is also used to access classes and sub-packages from the package.

### **Q.144)What’s the difference between the methods sleep() and wait()?**

| **Sleep()** | **Wait()** |
| --- | --- |
| The sleep() method belongs to the thread class. | Wait() method belongs to the object class. |
| Sleep does not release the lock that the current thread holds. | wait() release the lock which allows other threads to acquire it. |
| This method is a static method. | This method is not a static method. |
| Sleep() does not throw an InterruptedException. | InterruptedException is shown if the thread is interrupted while waiting. |
| Mainly used to delay a thread for some specific time duration. | Mainly used to pause a thread until notified by another thread. |

### **Q.145) Why is StringBuffer called mutable?**

StringBuffer class in Java is used to represent a changeable string of characters. It offers an alternative to the immutable String class by enabling you to change a string’s contents without constantly creating new objects. Mutable (modifiable) strings are created with the help of the StringBuffer class. The StringBuffer class in Java is identical to the String class except that it is changeable.

**Q.146)What are the types of an array?**

There are two types of arrays i.e., Primitive arrays and References Arrays.

* **Single-Dimensional Arrays:** Arrays that have only one dimension i.e., an array of integers or an array of strings are known as single-dimensional arrays.
* **Multi-Dimensional Arrays:** Arrays that have two or more dimensions such as two-dimensional or three-dimensional arrays.

### **Q.147) Explain various interfaces used in the Collection framework.**

Collection framework implements

1. Collection Interface
2. List Interface
3. Set Interface
4. Queue Interface
5. Deque Interface
6. Map Interface

Q.148)**When Abstract methods are used?**

An abstract method is used when we want to use a method but want to child classes to decide the implementation in that case we use Abstract methods with the parent classes.

### **Q.149)What is the difference between Comparable and Comparator?**

| **No.** | **Comparable** | **Comparator** |
| --- | --- | --- |
| 1) | Comparable provides only one sort of sequence. | The Comparator provides multiple sorts of sequences. |
| 2) | It provides one method named compareTo(). | It provides one method named compare(). |
| 3) | It is found in java.lang package. | It is located in java.util package. |
| 4) | If we implement the Comparable interface, The actual class is modified. | The actual class is not changed. |

**Q.150)What does the hashCode() method?**

The hashCode() method returns a hash code value (an integer number).