Interview Question

A What is JAVA?

Java is a high level programming language and platform independent. Java is a collection of objects.It was developed by sun Microsystem.

B What do you mean by Constructor?

Constructor is a method which name is same as class name and is used to initialize an object.

C What is meant by the Local variable and the Instance variable?

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

**Instance variable** is defined inside the class and outside the method and the scope of the variables exists throughout the class.

Q 4. 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.

Java has Bootstrap, Extension, and Application classloaders.

Q 5. What are the Memory Allocations available in JavaJava?

Java has five significant types of memory allocations.

* Class Memory
* Heap Memory
* Stack Memory
* Program Counter-Memory
* Native Method Stack Memory

Q 6. What are the differences between Heap and Stack Memory in Java?

[Stack](https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-in-data-structures) memory is the amount of memory allocated to each individual programme. It is a fixed memory space. Heap memory, in contrast, is the portion that was not assigned to the Java code but will be available for use by the Java code when it is required, which is generally during the program's runtime.

Q 7. Will the program run if we write static public void main?

Yes, the program will successfully execute if written so. Because, in Java, there is no specific rule for the order of specifiers

8. What is the default value stored in Local Variables?

Neither the Local Variables nor any primitives and Object references have any default value stored in them.

Q 9. Explain the expected output of the following code segment?

public class Simplilearn

{

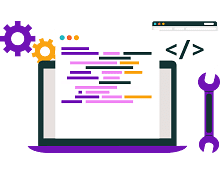
    public static void main (String args[])

    {

        System.out.println(100 + 100 +“Simplilearn");

        System.out.println(“E-Learning Company" + 100 + 100);

    }

}

Q 10. What is an Association?An Association can be defined as a relationship that has no ownership over another. For example, a person can be associated with multiple banks, and a bank can be related to various people, but no one can own the other.

Q 11. 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 12. Define Copy Constructor in Java

A Copy Constructor in Java is a constructor that initializes an object through another object of the same class.

Q 13. 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 14. 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 15. 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 16. What is an object-oriented paradigm?

A Paradigm that is based on the concepts of “Objects.” It contains data and code. Data that is in the form of fields, and regulation, that is in the form of procedures. The exciting feature of this paradigm is that the object’s procedures can access and often modify the data fields themselves.

Q 17. Define Wrapper Classes in Java.

In Java, when you declare primitive datatypes, then Wrapper classes are responsible for converting them into objects(Reference types).

Q 18. 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 19. Define package in Java.

The package is a collective bundle of classes and interfaces and the necessary libraries and JAR files. The use of packages helps in code reusability.

Q 20. Can you implement pointers in a Java Program?

Java Virtual Machine takes care of memory management implicitly. Java's primary motto was to keep programming simple. So, accessing memory directly through pointers is not a recommended action. Hence, pointers are eliminated in Java.

Q 21. Differentiate between instance and local variables.

For instance, [variables](https://www.simplilearn.com/tutorials/javascript-tutorial/scope-of-variables-in-javascript) are declared inside a class, and the scope is limited to only a specific object.A local variable can be anywhere inside a method or a specific block of code. Also, the scope is limited to the code segment where the variable is declared.

Q 22. 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 23. What is an Exception?An [Exception in Java](https://www.simplilearn.com/tutorials/java-tutorial/exception-handling-in-java) is considered an unexpected event that can disrupt the program's normal flow. These events can be fixed through the process of Exception Handling.

Q 24. 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 25. 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."

With this we are done with the first section that is Basic Java Interview Question, Now, lets move on to our next section of Intermediate Java Interview Questions.

Java Coding Interview Questions For Intermediate

Now, let's have a look at some of the most asked Java interview questions for intermediate experienced professionals.

Q 26. What is JDK? Mention the variants of JDK?

JDK is an abbreviation for Java Development Kit. It is a combined Package of JRE and Developer tools used for designing [Java Applications](https://www.simplilearn.com/popular-java-applications-article) and Applets. Oracle has the following variants.

* JDK Standard Edition
* JDK Enterprise Edition
* JDK Micro Edition
* Q 27. What is the difference between JDK, JRE, and JVM?

JVM has a Just in Time (JIT) compiler tool that converts all the Java source code into the low-level compatible machine language. Therefore, it runs faster than the regular application.

JRE has class libraries and other JVM supporting files. But it doesn’t have any tool for java development such as compiler or debugger.

JDK has tools that are required to write Java Programs and uses JRE to execute them. It has a compiler, Java application launcher, and an applet viewer.

Q 28. What is a JIT compiler?

JIT compiler refers to Just in Time compiler. It is the simplest way of executing the computer code that takes in compilation during the execution of a program rather than before performance. It commonly uses bytecode translation to machine code. It is then executed directly.

Q 29. What are Brief Access Specifiers and Types of Access Specifiers?

Access Specifiers are predefined keywords used to help JVM understand the scope of a variable, method, and class. We have four access specifiers.

* Public Access Specifier
* Private Access Specifier
* Protected Access Specifier
* Default Access Specifier

Q 30. How many types of constructors are used in Java?

There are two [types of constructors](https://www.simplilearn.com/tutorials/java-tutorial/constructor-in-java) that are used in Java.

Parameterized Constructors: Parameterized constructor accepts the parameters with which users can initialize the instance variables. Users can initialize the class variables dynamically at the time of instantiating the class.

Default constructors: This type doesn’t accept any parameters; rather, it instantiates the class variables with their default values. It is used mainly for object creation.

Q 31. Can a constructor return a value?

Yes, A constructor can return a value. It replaces the class's current instance implicitly; you cannot make a constructor return a value explicitly.

Q 32. Explain ‘this’ keyword in Java.

The term "this" is a particular keyword designated as a reference keyword. The "this" keyword is used to refer to the current class properties like method, instance, variable, and constructors.

Q 33. Explain ‘super’ keyword in Java.

The term "super" is a particular keyword designated as a reference keyword. The "super" keyword refers to the immediate parent class object.

Q 34. Explain Method Overloading in Java.

The process of creating multiple method signatures using one method name is called Method Overloading in Java. Two ways to achieve method overloading are:

1. Varying the number of arguments
2. Changing the return type of the Method

Q 35. Can we overload a static method?

No, Java does not support the Overloading of a [static method.](https://www.simplilearn.com/tutorials/java-tutorial/static-keyword-in-java) The process would throw an error reading "static method cannot be referenced."

Q 36. Define Late Binding.

Binding is a process of unifying the method call with the method's code segment. Late binding happens when the method's code segment is unknown until it is called during the runtime.

Q 37. Define Dynamic Method Dispatch.

The Dynamic method dispatch is a process where the method call is executed during the runtime. A reference variable is used to call the super-class. This process is also known as Run-Time Polymorphism.

Q 38. Why is the delete function faster in the linked list than an array?

Delete Function is faster in [linked lists](https://www.simplilearn.com/tutorials/java-tutorial/linked-list-in-java) as the user needs to make a minor update to the pointer value so that the node can point to the next successor in the list

Q 39. Give a briefing on the life cycle of a thread.

The life cycle of a thread includes five stages, as mentioned below.

1. New Born State
2. Runnable State
3. Running State
4. Blocked State
5. Dead State

Q 40. 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 41. Brief the life cycle of an applet.

The life cycle of an applet involves the following.

1. Initialization
2. Start
3. Stop
4. Destroy
5. Paint

Q 42. 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 43. Explain the Externalizable interface.

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

Q 44. 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 45. Explain the term enumeration in Java.

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

Q 46. 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 47. 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 48. 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.

Java Interview Questions for Experienced

Now, lets move on to our last section of Advanced Core Java Interview Questions which is primarly useful for experienced and working professionals.

Q 49. Can "this" and "super" keywords be used together?

No, "this" and "super" keywords should be used in the first statement in the class constructor. The following code gives you a brief idea.

public class baseClass {

     baseClass() {

         super();

         this();

         System.out.println(" baseClass object is created");

     }

     public static void main(String []args){

         baseClass bclass = new baseClass();

     }

}

Q 50. What is a JSP page?

JSP is an abbreviation for Java Servlet Page. The JSP page consists of two types of text.

* Static Data
* JSP elements

Q 51. 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 52. Explain the various directives in JSP.

Directives are instructions processed by JSP Engine. After the JSP page is compiled into a Servlet, Directives set page-level instructions, insert external files, and define customized tag libraries. Directives are defined using the symbols below:

start with "< %@" and then end with "% >"

The various types of directives are shown below:

* Include directive

It includes a file and combines the content of the whole file with the currently active pages.

* Page directive

Page Directive defines specific attributes in the JSP page, like the buffer and error page.

* Taglib

Taglib declares a custom tag library, which is used on the page.

Q 53. What are the observer and observable classes?

Objects that inherit the "Observable class" take care of a list of "observers."

When an Observable object gets upgraded, it calls the update() method of each of its observers.

After that, it notifies all the observers that there is a change of state.

The Observer interface gets implemented by objects that observe Observable objects.

Q 54. What is Session Management in Java?

A session is essentially defined as the random conversation's dynamic state between the client and the server. The virtual communication channel includes a string of responses and requests from both sides. The popular way of implementing session management is establishing a session ID in the client's communicative discourse and the server.

Q 55. Briefly explain the term Spring Framework.

Spring is essentially defined as an application [framework](https://www.simplilearn.com/tutorials/java-tutorial/java-frameworks) and inversion of control containers for Java. The spring framework creates enterprise applications in Java. Especially useful to keep in mind that the spring framework's central features are essentially conducive to any Java application.

Q 56. How to handle exceptions in Spring MVC Framework?

Spring MVC has two approaches for handling the exceptions:

* Exception handler method: In this kind of exception handling, the user will get the @ExceptionHandler annotation type used to annotate a method to handle exceptions.
* XML Configuration: The user can use the SimpleMappingExceptionResolver bean in Spring’s application file and map the exception.

Q 57. What is JCA in Java?

Java Cryptography Architecture gives a platform and provides architecture and application programming interfaces that enable decryption and encryption.

Developers use Java Cryptography Architecture to combine the application with the security applications. Java Cryptography Architecture helps in implementing third party security rules and regulations.

Java Cryptography Architecture uses the hash table, encryption message digest, etc. to implement the security.

Q 59. Explain the different authentications in Java Servlets.

Authentication options are available in Servlets: There are four different options for authentication in servlet:

* Basic Authentication:

Usernames and passwords are given by the client to authenticate the user.

* Form-based authentication:

In this, the login form is made by the programmer by using HTML.

* Digest Authentication:

It is similar to basic authentication, but the passwords are encrypted using the Hash formula. Hash Formula makes digest more secure.

* Client certificate Authentication:

It requires that each client accessing the resource has a certificate that it sends to authenticate itself. Client Authentication requires the SSL protocol.

Q 60. Explain FailFast iterator and FailSafe iterator along with examples for each.

FailFast iterators and FailSafe iterators are used in Java Collections.

FailFast iterators do not allow changes or modifications to the Java Collections, which means they fail when the latest element is added to the collection or an existing element gets removed from the collection. The FailFast iterators tend to fail and throw an exception called ConcurrentModificationException.

Ex: ArrayList, HashMap

Whereas, on the other hand, FailSafe iterators allow changes or modifications to be done on the Java Collections. It is possible, as the FailSafe iterators usually operate on the cloned copy of the collection. Hence, they do not throw any specific exception.

Ex: CopyOnWriteArrayList

Q 61. How do we reverse a string?

The [string can be reversed](https://www.simplilearn.com/tutorials/java-tutorial/reverse-a-string-in-java) by using the following program.

package simplilearnJava;

public class StringReverse {

public static void main(String args[]) {

String str = "Simplilearn";

String reverse = new StringBuffer(str).reverse().toString();

System.out.printf("Actual Word: %s, Word after reversing %s", str, reverse);

}

public static String reverse(String source) {

if (source == null || source.isEmpty()) {

return source;

}

String reverse = "";

for (int i = source.length() - 1; i >= 0; i--) {

reverse = reverse + source.charAt(i);

}

return reverse;

}

}

Expected Output:

Actual Word: Simplilearn, Word after reversing nraelilpmiS

Q 62. Write a program to find the square root of a number.

The Square root of a number can be found by using the following program.

package simplilearnJava;

import java.util.Scanner;

public class SRoot {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("Input a number to find square root: ");

double square = sc.nextDouble();

double squareRoot = Math.sqrt(square);

System.out.printf("The square root is: %f ", squareRoot);

}

}

}

Expected Output:

Input a number to find square root:

25

The square root is: 5

Q 63. Write a program that detects the duplicate characters in a string.

The program that finds the duplicate elements in a string is written below:

package simplilearnJava;

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

public class FindDuplicate {

public static void main(String args[]) {

printDuplicateCharacters("Simplilearn");

}

public static void printDuplicateCharacters(String word) {

char[] characters = word.toCharArray();

Map<Character, Integer> charMap = new HashMap<Character, Integer>();

for (Character ch : characters) {

if (charMap.containsKey(ch)) {

charMap.put(ch, charMap.get(ch) + 1);

} else {

charMap.put(ch, 1);

}

}

Set<Map.Entry<Character, Integer>> entrySet = charMap.entrySet();

System.out.printf("List of duplicate characters in String '%s' %n", word);

for (Map.Entry<Character, Integer> entry : entrySet) {

if (entry.getValue() > 1) {

System.out.printf("%s: %d %n", entry.getKey(), entry.getValue());

}

}

}

}

Expected output:

List of duplicate characters in String 'Simplilearn.'

i: 2

l: 2

Q 64. Write a Program to remove duplicates in an ArrayList.

The following program can be implemented to remove duplicate elements in an ArrayList

package simplilearnJava;

import java.util.ArrayList;

import java.util.LinkedHashSet;

import java.util.List;

import java.util.Set;

public class ArrayDuplicate {

public static void main(String args[]) {

List<Integer> num = new ArrayList<Integer>();

num.add(1);

num.add(2);

num.add(3);

num.add(4);

num.add(5);

num.add(6);

num.add(3);

num.add(4);

num.add(5);

num.add(6);

System.out.println("Your list of elements in ArrayList : " + num);

Set<Integer> primesWithoutDuplicates = new LinkedHashSet<Integer>(num);

num.clear();

num.addAll(primesWithoutDuplicates);

System.out.println("list of original numbers without duplication: " + num);

}

}

Expected Output:

Your list of elements in ArrayList : [1, 2, 3, 4, 5, 6, 3, 4, 5, 6]

list of original numbers without duplication: [1, 2, 3, 4, 5, 6]

Q 65. Find the word count in a string using HashMap Collection.

The following program can be used for word count.

package simplilearnJava;

import java.util.HashMap;

public class WordCount {

public static void main(String[] args) {

String str = "Hello World, Welcome to Simplilearn";

String[] split = str.split(" ");

HashMap<String, Integer> map = new HashMap<String, Integer>();

for (int i = 0; i < split.length; i++) {

if (map.containsKey(split[i])) {

int count = map.get(split[i]);

map.put(split[i], count + 1);

} else {

map.put(split[i], 1);

}

}

System.out.println(map);

}

}

Expected Output:

{Hello=1, Simplilearn=1, Welcome=1, to=1, World,=1}

Q 66. Write a program to find the Second Highest number in an ArrayList

The following program can be used to find the second biggest number in an array list.

package simplilearnJava;

public class NextHighest {

public static void main(String[] args)

    {

        int array[] = { 1, 2, 3, 4, 11, 12, 13, 14, 21, 22, 23, 24, 31, 32};

        int high = 0;

        int nextHigh = 0;

        System.out.println("The given array is:");

        for (int i = 0; i < array.length; i++)

        {

            System.out.print(array[i] + "\t");

        }

        for (int i = 0; i < array.length; i++)

        {

            if (array[i] > high)

            {

                nextHigh = high;

                high = array[i];

            }

            else if (array[i] > nextHigh)

            {

                nextHigh = array[i];

            }

        }

        System.out.println("Second Highest is:" + nextHigh);

        System.out.println("Highest Number is: "  +high);

    }

}

Expected Output:

The given array is:

1 2 3 4 11 12 13 14 21 22 23 24 31 32

Second Highest is:31

The highest number is: 32

Q 67. 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 68. Could you provide some implementation of a Dictionary having a large number of words?

The simplest implementation that can be given is that of a List wherein one can place ordered words and perform a Binary search. The other implementation with a better search performance is HashMap where the key is used as the first character of the word and the value as a LinkedList.

Up another level, there are HashMaps like:

hashmap {

a (key) -> hashmap (key-aa , value (hashmap(key-aaa,value)

b (key) -> hashmap (key-ba , value (hashmap(key-baa,value)

z (key) -> hashmap (key-za , value (hashmap(key-zaa,value)

}

Up to n levels where n is the average size of the word in the dictionary.

Q 69. How would you tackle it if you might have to encounter pattern programs in Java?

Solution - [Top 25 Most Frequently asked Pattern Programs in Java](https://www.simplilearn.com/tutorials/java-tutorial/pattern-programs-in-java)

With this, we have come to the end of this Java Interview Questions article. Moving ahead, we will look into the next crucial steps that you could pursue, to master Java.

Q 70. What do you understand by an instance variable and a local variable?

Generally, instance variables are declared in a class but outside methods whereas a local variable is declared within the blocks of code.

//Local Variable

import Java.io.\*;

class Main {

public static void main(String[] args)

{

int var = 145;

System.out.println("Local Variable: " + var);

}

}

//Instance variable

import Java.io.\*;

class Main {

public int value = 12;

public static void main(String[] args)

{

Main va = new Main();

System.out.println("My value is: " + va.value);

}

}

Q 75. When can you use the "super" keyword?

Basically, the super keyword is used to refer to the parent class. When there are the same fields in both parent and child classes, then one can use a super keyword to access data members of the parent class.

Q 76. What are shallow copy and deep copy in Java?

In the case of a shallow copy, primitive data types are copied, whereas in the case of a deep copy along with primitive data types the object references are also copied.

Q 77. Using relevant properties highlight the differences between interfaces and abstract classes.

An abstract class can have a combination of both abstract and non-abstract methods, whereas an interface has only abstract methods in it.

Q 78. 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.

Extending the Thread class

class InterviewBitThreadExample extends Thread{

   public void run(){

       System.out.println("Thread runs...");

   }

   public static void main(String args[]){

       InterviewBitThreadExample ib = new InterviewBitThreadExample();

       ib.start();

   }

}

Implementing the Runnable interface

class InterviewBitThreadExample implements Runnable{

   public void run(){

       System.out.println("Thread runs...");

   }

   public static void main(String args[]){

       Thread ib = new Thread(new InterviewBitThreadExample());

       ib.start();

   }

}

Implementing a thread using the method of Runnable interface is more preferred and advantageous as Java does not have support for multiple inheritances of classes.

start() method is used for creating a separate call stack for the thread execution. Once the call stack is created, JVM calls the run() method for executing the thread in that call stack.

Q 79. 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 80. Identify the output of the below Java program and Justify your answer.

class Main {

    public static void main(String args[]) {

        Scaler s = new Scaler(5);

    }

}

class InterviewBit{

    InterviewBit(){

        System.out.println(" Welcome to InterviewBit ");

    }

}

class Scaler extends InterviewBit{

    Scaler(){

        System.out.println(" Welcome to Scaler Academy ");

    }

    Scaler(int x){

        this();

        super();

        System.out.println(" Welcome to Scaler Academy 2");

    }

}

The above code will throw the compilation error. It is because the super() is used to call the parent class constructor. But there is the condition that super() must be the first statement in the block. Now in this case, if we replace this() with super() then also it will throw the compilation error. Because this() also has to be the first statement in the block. So in conclusion, we can say that we cannot use this() and super() keywords in the same block.

Q 81. Java works as a “pass by value” or “pass by reference” phenomenon?

Java works as a “pass by value” phenomenon, because “pass by reference” needs the help of pointers. But there are no pointers in Java.

Q 82. How to not allow serialization of attributes of a class in Java?

One approach to not allow serialization of attributes of a class in Java is by using writeObject() and readObject() methods in the subclass and throwing a not Serializable exception.

Q 83. What are the default values assigned to variables and instances in Java?

By default, for a numerical value it is 0, for the boolean value it is false and for objects it is NULL.

Q 84. What do you mean by data encapsulation?

Data encapsulation is one of the properties of OOPS concepts, where all the data such as variables and methods are enclosed together as a single unit.

Q 85. Can you tell the difference between equals() method and equality operator (==) in Java?

Equality operator (==) is used to check the equality condition between two variables. But the equals() method is used to check the equality condition between two objects.

Q 86. How is an infinite loop declared in Java?

An infinite loop can be declared in Java by breaking the logic in the instruction block.  For example,

for(int i = 1; i > 0; i++)

{

//statements

}

The above code forms an infinite loop in Java.

Q 87. Briefly explain the concept of constructor overloading

The concept of constructor overloading refers to having multiple methods in a class with their name being the same as the class name. The difference lies in the set of parameters passed to the functions.

Q 88. Explain the use of the final keyword in variable, method and class.

In Java, one can apply the final keyword to a variable, methods, and class. With the help of the final keyword, the variable turns out to be a constant, the method cannot be inherited and the class cannot be overridden.

Q 89. Is it possible that the ‘finally' block will not be executed? If yes then list the case.

Yes, there is a possibility that the ‘finally’ block cannot get executed. Here are some of the cases where the above situation occurs.

1. During the time of fatal errors such as memory exhaustion, memory access error, etc.
2. During the time of using System.exit()

Q 90. 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 91. What is the main objective of garbage collection?

The main goal of using garbage collection is to free the heap memory by eliminating unnecessary objects.

Q 92. Apart from the security aspect, what are the reasons behind making strings immutable in Java?

Because of security, synchronization, concurrency, caching, and class loading, the String is immutable in Java. The reason for making string final would be to destroy its immutability and help stop others from trying to extend it. String objects are cached in the String pool, making them immutable.

Q 93. Which of the below generates a compile-time error? State the reason.

int[] n1 = new int[0];

boolean[] n2 = new boolean[-200];

double[] n3 = new double[2241423798];

char[] ch = new char[20];

We get a compile-time error in line 3. The error we will get in Line 3 is - the integer number too large. It is because the array requires size as an integer.  And Integer takes 4 Bytes in the memory. And the number (2241423798) is beyond the capacity of the integer. The maximum array size we can declare is - (2147483647).

Because the array requires the size in integer, none of the lines (1, 2, and 4) will give a compile-time error. The program will compile fine. But we get the runtime exception in line 2. The exception is - NegativeArraySizeException.

Here what will happen is - At the time when JVM will allocate the required memory during runtime then it will find that the size is negative. And the array size can’t be negative. So the JVM will throw the exception.

Q 94. 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 95. What is a Comparator in Java?

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

Q 95. In Java, static as well as private method overriding is possible. Comment on the statement.

In Java, you could indeed override a private or static method. If you create a similar method in a child class with the same return type and method arguments, it will hide the super class method; this is known as method hiding. Similarly, you cannot override a private method in a subclass because it is not accessible from that.

Q 96. 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 97. Why is the character array preferred over string for storing confidential information?

Because Strings are immutable, any change will result in the creation of a new String, whereas char[] allows you to set all of the elements to blank or zero. So storing a password in a character array clearly reduces the security risk of password theft.

Q 98. What are the differences between HashMap and HashTable in Java?

|  |  |
| --- | --- |
| HashMap | HashTable |
| 1. Asynchronous in nature | 1. Synchronous in nature |
| 1. Not thread-safe | 2. Thread safe |
| 1. It allows one null key and null values | 3. It doesn’t allow null keys and values. |

Q 99. What is the importance of reflection in Java?

Reflection is a property of Java, enabling the Java code to inspect itself. A Java class, for example, can get the names of all its members and showcase them.

Q 100. What are the different types of Thread Priorities in Java? And what is the default priority of a thread assigned by JVM?

There are different types of thread properties in Java. They are MIN\_PRIORITY, MAX\_PRIORITY, and NORM\_PRIORITY. By default, the thread is assigned NORM\_PRIORITY.

Q 101. 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.