Top 70 Core Java Interview Questions Answered for Freshers and Experienced in 2022

By Ravikiran A S

Last updated on Apr 19, 2022

Table of Contents

Java is the most widely used programming language in the current IT industry. One major reason for the vast number of beginners and professionals in the field of programming is the career potential that Java knowledge comes with. This article is dedicated to the same purpose. Here is a complete guide on how to help you crack the most frequently asked Java Interview questions.

Basic Java Interview Questions for Freshers

So let's get started with the first set of basic Java Interview Questions which is primarly useful for freshers.

1. What are the differences between C++ and Java?

- **Concept.**

C++ is not platform-independent; the principle behind C++ programming is "write once, compile anywhere."

In contrast, because the byte code generated by the Java compiler is platform-independent, it can run on any machine, Java programs are written once and run everywhere.

- **Languages Compatibility.**

C++ is a programming language that is based on the C programming language. Most other high-level languages are compatible with C++.

Most of the languages of Java are incompatible. Java is comparable to those of C and C++.

- **Interaction with the library.**

It can access the native system libraries directly in C++. As a result, it's better for programming at the system level.

Java's native libraries do not provide direct call support. You can use Java Native Interface or access the libraries.

- **Characteristics.**

C++ distinguishes itself by having features that are similar to procedural and object-oriented languages. The characteristic that sets Java apart is automatic garbage collection. Java doesn't support destructors at the moment.

- **The semantics of the type.**

Primitive and object types in C++ have the same kind of semantics. The primitive and object classes of Java, on the other hand, are not consistent.

- **In the context of Compiler and Interpreter.**

Java refers to a compiled and interpreted language. In contrast, C++ is only a compiled language.

In Java, the source code is the compiled output is a platform-independent byte code.

In C++, the source program is compiled into an object code that is further executed to produce an output.

2. List the features of the Java Programming language?

A few of the significant features of Java Programming Language are:

Easy: Java is a language that is considered easy to learn. One fundamental concept of OOP Java has a catch to understand.

Secured Feature: Java has a secured feature that helps develop a virus-free and tamper-free system for the users.

OOP: OOP stands for Object-Oriented Programming language. OOP signifies that, in Java, everything is considered an object.

Independent Platform: Java is not compiled into a platform-specific machine; instead, it is compiled into platform-independent bytecode. This code is interpreted by the Virtual Machine on which the platform runs.

3. What do you get in the Java download file? How do they differ from one another?

We get two major things along with the Java Download file.

JDK - Java Development Kit

JRE - Java Runtime Environment

JDK	JRE
Abbreviation for JavaDevelopment Kit	Abbreviation for Java Runtime Environment
JDK is a dedicated kit for solely software development	JRE is a set of software and library designed for executing Java Programs

Unlike JVM, JDK is Platform Dependent	Unlike JVM, JRE is also Platform Dependent
JDK package is a set of tools for debugging and Developing	JRE Package is one that only supports files and libraries for a runtime environment
JDK package will be provided with an installer file	JRE Package does not get an installer but has only a runtime environment

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.

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

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

Stack is generally used to store the order of method execution and local variables. In contrast, Heap memory is used to store the objects. After storing, they use dynamic memory allocation and deallocation.

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.

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);
   }
}
```

The answers for the two print statements are as follows.

- 200Simplilearn
- E-Learning Company100100

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.

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.

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.

13. What is a Marker Interface?

An empty interface in Java is referred to as a Marker interface. Serializable and Cloneable are some famous examples of Marker Interface.

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.

15. Why is Java not completely object-oriented?

Java is not considered as a 100% object-oriented programming language because it still makes use of eight or more primitive data types like int, float double, etc.

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.

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).

18. Define Singleton Classes in Java.

In Java, when you make the constructor of a class private, that particular class can generate only one object. This type of class is popularly known as a Singleton Class.

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.

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.

21. Differentiate between instance and local variables.

For instance, variables 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.

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.

23. What is an Exception?

An Exception 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.

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.

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.

Intermediate Java Interview Questions

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

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 and Applets. Oracle has the following variants.

- JDK Standard Edition
- JDK Enterprise Edition
- JDK Micro Edition

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.

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.

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

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

There are two types of constructors 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.

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.

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.

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.

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

35. Can we overload a static method?

No, Java does not support the Overloading of a static method. The process would throw an error reading "static method cannot be referenced."

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.

37. Define Dynamic Method Dispatch.

Previous Next

Tutorial Playlist

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

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

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

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.

43. Explain the Externalizable interface.

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

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.

45. Explain the term enumeration in Java.

Enumeration or enum is an interface in Java. Enum allows the sequential access of the elements stored in a collection in Java.

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.

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.

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.

Advanced Java Interview Questions for Experienced

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

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();
    }
}
```

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

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

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.

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.

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.

55. Briefly explain the term Spring Framework.

Spring is essentially defined as an application framework 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.

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.

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.

58. Explain JPA in Java.

The Java Persistence API enables us to create the persistence layer for desktop and web applications. Java Persistence deals in the following:

1. Java Persistence API
2. Query Language
3. Java Persistence Criteria API
4. Object Mapping Metadata

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.

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

61. How do we reverse a string?

The string can be reversed 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

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

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
```

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]
```

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}
```

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];
```

```
            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

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.

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.

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

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.

Next Steps

Java Interview Questions are really important to go through before attending an interview. It helps you to put yourself on the safer side by getting you ready to be able to answer the questions asked in your interview. Now, the next step is to learn How to Become a Software Developer.

You can also explore and get familiar with interview questions related to other backend languages like PHP, Python and Node.js Interview Questions.

If you're looking for more in-depth knowledge about the Java programming language and information on how to get certified as a professional developer, explore our Java training and certification programs, which Simplilearn's experienced industry experts offer in real-time. In particular, check out our Full Stack Java Developer Master's Program today!

If you have any questions about this "Java Interview Questions" article, please leave them in the comments section, and our experts will answer them for you, at the earliest!

Find our Full Stack Java Developer Online Bootcamp in top cities:

Name	Date	Place	
Full Stack Java Developer	Cohort starts on 27th Apr 2022, Weekend batch	Your City	View Deta
Full Stack Java Developer	Cohort starts on 29th Apr 2022, Weekend batch	Hyderabad	View Deta
Full Stack Java Developer	Cohort starts on 11th May 2022, Weekend batch	Pune	View Deta

About the Author

Ravikiran A S

Ravikiran A S works with Simplilearn as a Research Analyst. He an enthusiastic geek always in the hunt to learn the latest technologies. He is proficient with Java Programming Language, Bi...

View More

Recommended Resources

Implementing Stacks in Data Structures

Blockchain Career Guide: A Comprehensiv...

Name

MURALI • a month ago
VERY VERY HELPFUL... THANKS A LOT ... GREAT JOB
Reply • Share ›

Team Simplilearn Mod ➚ MURALI • 23 days ago
Hi Murali, glad you found it helpful.
Reply • Share ›

Subba Rao Kesa • 3 months ago
35. Can we overload a static method?
Yes
Reply • Share ›

Team Simplilearn Mod ➚ Subba Rao Kesa • 2 months ago
Appreciate your feedback.
1 Reply • Share ›

Sandra livesay • 4 months ago
Thanks a million for providing answers for 70 Core Java Interview Questions.
Reply • Share ›

Team Simplilearn Mod ➚ Sandra livesay • 4 months ago
Hi Sandra, glad you found it helpful.
Reply • Share ›

Rehaan • 4 months ago
Well written article on java interview questions. Must know!
Reply • Share ›

Team Simplilearn Mod ➚ Rehaan • 4 months ago
Hi Rehaan, glad you liked it.
Reply • Share ›

Harsha V • 4 months ago
Since I am new to Java, I didn't know what type of questions would I get in the interview. These **Java Interview questions and answers** article help me to get clear understanding on both questions and answers.
Reply • Share ›

Team Simplilearn Mod ➚ Harsha V • 4 months ago
Hi Harsha, glad you found it helpful.
Reply • Share ›