By OnlineInterviewQuestions.com

HCL Interview Questions for Java Developer

Q1. What is Multithreading in Java?

Multithreading is a programmable approach to achieve multitasking. Multithreading in Java is a process of executing multiple threads cumulatively. A thread is the smallest unit of processing which is a lightweight sub-process. It is a time-saving approach.

Q2. What use of intern() method in String?

In Java, intern() is a native method of the String class. The String.intern () returns a reference to the equal string literal present in the string pool. The intern () method is applicable to String objects created via the 'new' keyword. When it is executed then it checks whether the String equals, this String object in the pool or not.

Q3. What is Collection Framework?

A Java collection framework is an architecture that was added to Java 1.2 version. The Collection Framework provides a well-designed set of classes and interfaces for storing and manipulating a group of data or objects as a single unit.

Q4. What are different types of Classes available in Java?

Different types of Classes available in Java are as follows:

1. Concrete class
- Abstract class
- Final class
- Static class
- Inner Class

Concrete class

In Java, A concrete class is a type of subclass, which implements all the abstract methods of its super abstract class from which it extends.

Abstract class

In Java, An abstract class is a type of class. An abstract class is declared by the abstract keyword. An abstract class cannot be instantiated directly. An abstract class can be instantiated either by a concrete subclass or by defining all the abstract methods along with the single statement that has no definition or body. It may or may not contain an abstract method. If a class contains an abstract method, then it also needs to be an abstract method.

Final class

A class declared with the final keyword is a final class and it cannot be extended by another class.

Static class

Static classes in Java are allowed only for inner classes which are defined under some other class,as static outer class is not allowed which means that we can't use static keywords with outer class.

Inner Class

A class declared within another class or method is called an inner class.

Q5. What are access modifiers in Java?

The four types access modifiers in Java are as follows:

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

private indicates that The access level of a private modifier is only within the class.

default indicates that The access level of a default modifier is only within the package.

protected indicates that The access level of a protected modifier is within the package and outside the package through child class.

public indicates that The access level of a public modifier is everywhere.

Q6. Explain string buffer in Java?

In Java, the StringBuffer class is used to create mutable (modifiable) strings. Java StringBuffer class is thread-safe i.e. multiple threads cannot access it simultaneously. So it is safe and will result in an order.

Q7. Write a program to Reverse a string in Java?

The JAVA Program to Reverse a string is as follows:

```
import java.io.*;
import java.util.*;
public class Reverse_String {
    public static void main(String[] args) {
        String input_string = "Conax";
        char[] try1 = input_string.toCharArray();
        for (int i = try1.length-1;i >= 0; i--) System.out.print(try1[i]);
    }
}
```

Q8. Tell me different ways to create an object in Java?

The object is a real world entity which has state, behavior and identity. You can say that object is an instance of a class.

Different ways to create an object in Java are as follows:

Using new keyword:

ClassName obj1 = new ClassName();

Using predefined Class class's newInstance() method:

ClassName obj2 = ClassName.class.newInstance();

Using Constructor class's newInstance() method:

Constructor constructor = ClassName.class.getConstructor();

ClassName obj3 = constructor.newInstance();

Using clone() method:

ClassName obj4 = (ClassName) obj3.clone();

Using deserialization:

ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("data.obj"));

out.writeObject(obj4);

ObjectInputStream in = new ObjectInputStream(new FileInputStream("data.obj"));

ClassName obj5 = (ClassName) in.readObject();

Q9. What are generics in Java?

The **Java Generics** programming is included in J2SE 5 to administer with type-safe things. Generics in Java are related to templates in C++. The purpose is to concede class (String, Integer, and user-defined sorts) to denote a parameter to designs, types, and interfaces. For instance, states like HashMap, HashSet, ArrayList, etc utilize generics pretty well. We may utilize them for each kind. We may write a design/class/interface already and apply for all kind we require. Before Generics, we needed to typecast.

Q10. Explain use of Lambda expressions in Java?

Java included a new feature in Java SE 8 which was termed as Lambda expression. It provides a clear and concise way. It is used to represent one method interface with the help of an expression. It is very useful in the collection library that helps to iterate, filter, and extract data from the collection. It is also used to provide an implementation of an interface that decreases the lines of code.

Q11. Write program to remove consecutive duplicate characters in a string?

A Program to remove consecutive duplicate characters in a string is as follows:

```
// Recursive Program to remove consecutive
// duplicates from string S.
#include <bits/stdc++.h>
using namespace std;

// A recursive function that removes
// consecutive duplicates from string S
void removeDuplicates(char* S)
{
    // When string is empty, return
    if (S[0] == '\0')
        return;

    // if the adjacent characters are same
    if (S[0] == S[1]) {

        // Shift character by one to left
        int i = 0;
        while (S[i] != '\0') {
            S[i] = S[i + 1];
            i++;
        }

        // Check on Updated String S
        removeDuplicates(S);
    }
```

```
    // If the adjacent characters are not same
    // Check from S+1 string address
    removeDuplicates(S + 1);
}

// Driver Program
int main()
{
    char S1[] = "geeksforgeeks";
    removeDuplicates(S1);
    cout << S1 << endl;

    char S2[] = "aabcca";
    removeDuplicates(S2);
    cout << S2 << endl;

    return 0; }
```

Output:

geksforgeks

abca

Q12. List few difference between overloading and overriding?

Few difference between Overloading and Overriding are as follows:

Overloading	Overriding
Overloading is the situation when two or more methods in one class have the same method name but different parameters.	Overriding is the situation that means having two methods with the same method name and parameters (i.e., method signature). One of the methods is in the parent class and the other is in the child class.
The real object type in the run-time, not the reference variable's type, determines which overridden method is used at runtime.	Overriding allows a child class to provide a specific implementation of a method that is already provided by its parent class. In contrast, reference type determines which overloaded method will be used at compile time.
Polymorphism applies to overriding.	Polymorphism does not apply to overloading.
Overriding is a concept of run-time.	Overloading is a concept of compile-time.

Q13. What is string pool?

In JAVA, The string pool is a memory that resides in the heap area of the JVM(Java virtual machine). Heap is a kind of tree data structure. It is specifically used to store objects in Java. In other words, a string pool is a pool of string stored in JAVA heap memory. A String is a special class in java and we can create String objects using a new operator.

Q14. What is Spring MVC?

Spring MVC framework is a type of **Model-View-Controller** architecture that provides components to develop flexible and loosely coupled web applications. This Java framework provides the DispatchServlet class to provide an easy solution to use MVC in the spring framework.

Q15. Explain Upcasting and downcasting in Java?

Upcasting and Downcasting are important parts of Java that allow us to build complicated programs using a simple syntax. Java permits an object of a subclass type to be treated as an object of any superclass type that is called upcasting. Downcasting can fail if the actual object type is not the target object type whereas Upcasting can not fail. Let's see an example: Here, we cast the subject type to the Science type. As Science is a subclass of the Subject, this casting is called Downcasting.

Q16. What are Executors in Java?

Executors is a class that provides ExecutorService that has different methods like newCachedThreadPool (), newScheduledThreadPool (), and newCachedThreadPool () that serves a different purpose to create a thread pool. java.util.concurrent.Executors is a package that has been introduced in JDK 5 and is part of the executor framework.

Q17. What is serialization in Java?

Serialization appears to be a device of changing the nature of an item within a byte current. Object serialization implies a method utilized to change the nature of an item into a byte stream, that may be continued into file or disk or transmitted over the chain to any other operating Java pragmatic machine. Also, to move the state of an object across a chain or to persist or save the nature of an item serialization is necessary.

Moreover, Serialization has no method and data member. It is mainly used to mark the Java classes so the object used in these classes gets its capabilities. Therefore, Serialization must always be implemented for whose object of the class you want to persist.

Q18. What are class loaders?

ClassLoader class has a method name defineClass which takes input as a byte array and loads a class. All class loaders except the bootstrap class loader have a parent class loader. Every class loader instance is associated with a parent class loader.

Q19. List different hibernate object type?

Different hibernate object types are listed below:
1. Entities and values
2. Basic value types
3. Custom value types

Q20. What are checked and unchecked exceptions in Java?

An Exception is a run time error especially occurring at execution time.

Following are the two types of Exceptions in JAVA:

1. Checked Exception
2. Unchecked Exception

In Java, The exceptions that are checked during the compilation time are termed as Checked exceptions to verify that a method that is throwing an exception contains the code to handle the exception with the try-catch block or not. if there is no code to handle them, then the compiler checks whether the method is declared using the throws keyword. And, if the compiler finds neither of the two cases, then it gives a compilation error.

A checked exception extends the Exception class. For instance, if we write a program to read data from a file using a FileReader class and if the file does not exist, then there is a FileNotFoundException.

An exception that checks during the execution of a program is called an unchecked or a runtime exception. The unchecked exceptions occur when the user attempts to access an element with an invalid index, calling the method with illegal arguments, etc. Unchecked exceptions are checked during the runtime. Therefore, the compiler does not check whether the user program contains the code to handle them or not.

For instance, if a program attempts to divide a number by zero. Or, when there is an illegal arithmetic operation. Suppose, we declare an array of size 10 in a program, and try to access the 12th element of the array, or with a negative index like -5, then we get an ArrayIndexOutOfBounds exception.

Q21. Write a program to remove consecutive duplicate characters in a string?

A Java program to remove consecutive duplicate characters in a string is as follows:

```
public class Hello
{
    public static void main(String[] args)
 {
        String str1 = "Online interviews";
        System.out.println(removeDuplicateChars(str1));
     }
```

```java
    private static String removeDuplicateChars(String sourceStr)
  {
        // Store encountered letters in this string.
        char[] chrArray = sourceStr.toCharArray();
        String targetStr = "";

        // Loop over each character.
        for (char value : chrArray)
  {
            // See if character is in the target
           if (targetStr.indexOf(value) == -1)
{
                targetStr += value;
// Use StringBuilder as shown below
            }
          }
        return targetStr;
     }
}
```

Output :

Onlie trvw

You could use StringBuilder as shown below :<?p>

```java
private static String removeDuplicates(String str)
{
    StringBuilder sb = new StringBuilder();
    char[] arr = str.toCharArray();
    for (char ch : arr)
{
        if (sb.indexOf(String.valueOf(ch)) != -1)
        continue;
        else
        sb.append(ch);
     }
    return sb.toString();
}
```

Q22. List a few differences between overloading and overriding?

Few differences between overloading and overriding are as follows:

S. N.	Method overloading	Method overriding
(1)	In this, methods must have the same name and different signatures.	While in this, methods must have the same name and same signature.
(2)	Method overloading is performed within the class.	Method overriding occurs in two classes that have IS-A (inheritance) relationships.
(3)	In case of method overloading, parameters must be different.	In case of method overriding, parameters must be the same.

S. N.	Method overloading	Method overriding
(4)	Method overloading is an example of compile-time polymorphism.	Method overriding is an example of run-time polymorphism.

Please Visit OnlineInterviewquestions.com to download more pdfs