Q1. How can you handle exceptions in Java?

Exception handling can be performed using:

Try: the set of statements or code which requires monitoring for an exception is kept under this block.

Catch: this block catches all exceptions that were trapped in the try block.

Finally: this block is always performed irrespective of the catching of exceptions in the try or catch block.

Q2. What is the difference between exception and error in Java?

Errors typically happen while an application is running. For instance, Out of Memory Error occurs in case the JVM runs out of memory. On the other hand, exceptions are mainly caused by the application. For instance, Null Pointer Exception happens when an app tries to get through a null object.

Q3. Why do we need exception handling in Java?

If there is no try and catch block while an exception occurs, the program will terminate. Exception handling ensures the smooth running of a program without program termination.

Q4. Name the different types of exceptions in Java

Based on handling by JVM, there are typically two types of exceptions in Java:

Checked: Occur during the compilation. Here, the compiler checks whether the exception is handled and throws an error accordingly.

Unchecked: Occur during program execution. These are not detectable during the compilation process.

In addition, there are two other exceptions based on their definition, namely built-in expectation and user-defined expectations.

Q5.What are run time exceptions in Java. Give example?

The exceptions which occur at run time are called as run time exceptions. These exceptions are unknown to compiler. All sub classes of java.lang.RunTimeException and java.lang.Error are run time exceptions. These exceptions are unchecked type of exceptions. For example, NumberFormatException, NullPointerException, ClassCastException, ArrayIndexOutOfBoundException, StackOverflowError etc.

Q6. Try-with-resources blocks? Why do we use them? When they are introduced?

Try-with-resources blocks are introduced from Java 7 to auto-close the resources like File I/O streams, Database connection, network connection etc… used in the try block. You need not to close the resources explicitly in your code. Try-with-resources implicitly closes all the resources used in the try block.

Benefits of try-with-resources?

The main benefit of try-with-resources is that it avoids resource leaks that could happen if we don’t close the resources properly after they are used. Another benefit of try-with-resources is that it removes redundant statements in the code and thus improves the readability of the code.

Q7. What is the difference between final, finally and finalize in java?

final keyword:

By declaring a variable as final, the value of final variable cannot be changed.

By declaring a method as final, method cannot be overridden.

By declaring a class as final, class cannot be extended.

finally:

Used after try or try-catch block, will get executed after the try and catch blocks without considering whether an exception is thrown or not.

finalize:

Finalize method is the method that Garbage Collector always calls just before the deletion/destroying the object which is no longer in use in the code.

Q8. What is RuntimeException in java. Give example?

The exceptions which occur at runtime are called as RuntimeException. These exceptions are unknown to the compiler. All subclasses of java.lang.RuntimeException are RuntimeExceptions.

For example:

NumberFormatException, NullPointerException, ClassCastException, ArrayIndexOutOfBoundException etc.

Q9. What is the difference between ClassNotFoundException and NoClassDefFoundError in java?

This question is important because very few Java developers are aware of the difference between ClassNotFoundException and NoClassDefFoundError.

ClassNotFoundException:

An exception that occurs when you try to load a class at run time using Class.forName() or loadClass() methods and mentioned classes are not found in the classpath is called ClassNotFoundException.

NoClassDefFoundError:

An exception that occurs when a particular class is present at compile-time but was missing at run time is called NoClassDefFoundError.

Q10. What is the difference between throws and throw in java?

This is one of the most frequently asked interview questions for java developers.

Main differences between throws and throw are :

a. throws keyword is used when writing methods, to declare that the method in question throws the specified (checked) exception.

throw is used when an instruction is to explicitly throw the exception.

b. throws is used with a method signature while the throw is used inside a method.