**Exception Interview Questions**

**Q. What is Exception Handling?**

**Ans.** Exception Handling in Java

The exception handling is one of the powerful mechanism provided in java. It provides the mechanism to handle the runtime errors so that normal flow of the application can be maintained.

In this page, we will know about exception, its type and the difference between checked and unchecked exceptions.

● Dictionary Meaning:Exception is an abnormal condition.

● In java, exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Exception Handling is a mechanism to handle runtime errors.

**Types of Exception:**

There are mainly two types of exceptions: checked and unchecked where error is considered as unchecked exception. The sun microsystem says there are three types of exceptions:

1. Checked Exception

2. Unchecked Exception

3. Error

**Q. What is difference between Checked Exception and Unchecked Exception?**

**Ans.**

**1)Checked Exception**

The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g.IOException, SQLException etc. Checked exceptions are checked at compile-time.

**2)Unchecked Exception**

The classes that extend RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time rather they are checked at runtime.

**3)Error**

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

**Q. What is the catch or declare rule for method declarations?**

**Ans.** If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

**Q. Can an exception be rethrown?**

**Ans.** Yes, an exception can be rethrown.

**Q. When is the finally clause of a try-catch-finally statement executed?**

**Ans.** The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

**Q. What class of exceptions are generated by the Java run-time system?**

**Ans.** The Java runtime system generates RuntimeException and Error exceptions.

**Q. What is the relationship between a method's throws clause and the exceptions that can be thrown during the method's execution?**

**Ans.** A method's throws clause must declare any checked exceptions that are not caught within the body of the method.

**Q. What classes of exceptions may be thrown by a throw statement?**

**Ans.** A throw statement may throw any expression that may be assigned to the Throwable type.

**Q. Which arithmetic operations can result in the throwing of an ArithmeticException?**

**Ans.** Integer / and % can result in the throwing of an ArithmeticException.

**Q. What happens if a try-catch-finally statement does not have a catch clause to handle an**

**exception that is thrown within the body of the try statement?**

**Ans.** The exception propagates up to the next higher level try-catch statement (if any) or results in the program's termination.

**Q. How does a try statement determine which catch clause should be used to handle an exception?**

**Ans.** When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exceptions executed. The remaining catch clauses are ignored.

**Q. Can an unreachable object become reachable again?**

**Ans.** An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

**Q. What is the base class for Error and Exception?**

**Ans.** Throwable

**Q. Is it necessary that each try block must be followed by a catch block?**

**Ans**. It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

**Q. What is finally block?**

**Ans.** The finally block is a block that is always executed. It is mainly used to perform some important tasks such as closing connection, stream etc.

Note:Before terminating the program, JVM executes finally block(if any). Note:finally must be followed by try or catch block.

**Q. Is there any case when finally will not be executed?**

**Ans.** Finally block will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

**Q. Can subclass overriding method declare an exception if parent class method doesn't throw an exception ?**

**Ans.** Yes but only unchecked exception not checked.

**Q. What is exception propagation ?**

**Ans.** Forwarding the exception object to the invoking method is known as exception propagation.