**What is exception?**

Exception is an abnormal condition which occurs during execution of a program and disrupts the normal flow of a program.

**Most Frequently Occurring Exceptions**

1.java.lang.NullPointerException:-occurs when your application tries to access null objects.

2.java.lang.ArrayIndexOutOfBoundsException:-occurs when you try to access an array element with an invalid index eg. index greater than the array length or with a negative index

3.java.lang.NumberFormatException:-this exception thrown when you are trying to convert a string to numeric numbers value like integer,float,double etc. but input string is not in numbers

4.java.lang.ClassNotFoundException:- this exception thrown when an application tries to load class at run time but the class with specified name is not found in the classpath

5.java.lang.ArithmeticException:- this exception thrown when an abnormal arithmetic condition arises in an application

6.java.sql.SQLException:- this exception thrown when an application encounters with an error while interacting with the database

7.java.lang.ClassCastException:- this exception occurs when an object of one type can not be casted to another type

8.java.io.IOException:- this exception occurs when an IO operation fails in your application

9.java.lang.StackOverflowError:-is a run time error which occurs when stack overflows. This happens when you keep calling the methods recursively

10.java.lang.NoClassDefFoundError:-this exception is thrown when java Runtime System tries to load the definition of a class which is no longer available.

11.java.lang.InterruptedException:- This exception is thrown when a thread is interrupted while it is in a blocked state, waiting for a resource or sleeping. Interruption is a way to signal a thread that it should stop what it's doing and exit gracefully.

12.java.lang.SecurityException:- This exception is thrown by the security manager to indicate a security violation. It typically occurs when a piece of code tries to perform an action that is not allowed due to security restrictions imposed by the security manager.

**Exception Handling In Java :**

Exceptions in Java are handled using try, catch and finally blocks.

try, catch and finally blocks form one unit. There must be one try block and one or more catch blocks. finally block is optional.There should not be any statements in between the blocks.

* try block : This block contains statements which may throw exceptions during run time.
* catch Block : This block handles the exceptions thrown by the try block.
* finally Block : This block is always executed whether an exception is thrown or not and thrown exception is caught or not. Hence, important code like releasing the resources are kept in this block.

Rules to follow While writing try-catch-finally blocks:-

1.try-catch and finally blocks form one unit. there must be one try block and one or more catch block. finally blocks is optional

2.there should not be any statements in between the blocks

3.if there are multiple catch blocks, the order of catch blocks must be from most specific to general ones. Eg. Lower classes in the hierarchy of exceptions must come first and higher classes later.

If try-catch-finally blocks are returning a value:-

1.if finally block return a value then try and catch blocks may or may not return a value.

2.if finally block does not return a value then both try and catch blocks must return a value.

3.Finally block overrides return values from try and catch blocks.

4.finally block will be always executed even though try and catch blocks are returning the control.

Types Of Exceptions :-There are two types of exceptions in Java.

Checked Exceptions:-are the exceptions which are checked during compilation itself. Checked exceptions are the exceptions which are checked during compilation itself. They are also called compile time exceptions. Compiler is aware of these exceptions and immediately throws the error wherever it sees the statements which may throw checked exceptions.

Unchecked Exceptions:-are the exceptions which are not checked during compilation.they occur only at run time. Unchecked exceptions are the exceptions which are not checked during compilation. Compiler is not aware of these exceptions. These exceptions occur only at run time. That’s why they are also called Run Time Exceptions.

|  |  |
| --- | --- |
| **Checked Exceptions** | **Unchecked Exceptions** |
| They are known at compile time. | They are known at run time. |
| They are checked at compile time. | They are not checked at compile time. |
| They are compile time exceptions. | They are run time exceptions. |
| These exceptions must be handled properly either using try-catch blocks or using throws clause, otherwise compiler will throw error. | If these exceptions are not handled properly, compiler will not throw any error. But, you may get error at run time. |
| All the sub classes of java.lang.Exception (except sub classes of java.lang.RunTimeException) are checked exceptions. | All the sub classes of java.lang.RunTimeException are unchecked exceptions. |
| Ex : FileNotFoundException, IOException, SQLException, ClassNotFoundException | Ex : NullPointerException, ArithmeticException, ClassCastException, ArrayIndexOutOfBoundsException |

**Hierarchy Of Exceptions :**

java.lang.Throwable is the super class for all type of errors and exceptions in Java.It has two sub classes.

java.lang.Error : It is the super class for all types of errors in Java.

java.lang.Exception : It is the super class for all types of exceptions in Java.

**throw Keyword :**

throw keyword is used to throw an exception explicitly.

where, InstanceOfThrowableType must be an object of type Throwable or subclass of Throwable.