**What is an exception?**

Exception is an abnormal condition which occurs during the execution of a program and disrupts normal flow of the program. This exception must be handled properly. If it is not handled, program will be terminated abruptly.

**2) How the exceptions are handled in java? OR Explain exception handling mechanism in java?**

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

try block : The code or set of statements which are to be monitored for exception are kept in this block.

catch block : This block catches the exceptions occurred in the try block.

finally block : This block is always executed whether exception is occurred in the try block or not and occurred exception is caught in the catch block or not.

**3) What is the difference between error and exception in java?**

Errors are mainly caused by the environment in which an application is running. For example, OutOfMemoryError happens when JVM runs out of memory. Where as exceptions are mainly caused by the application itself. For example, NullPointerException occurs when an application tries to access null object.

Click [here](https://javaconceptoftheday.com/difference-between-error-vs-exception-in-java/) to see more about Error Vs Exception in java.

**4) Can we keep other statements in between try, catch and finally blocks?**

No. We shouldn’t write any other statements in between try, catch and finally blocks. They form a one unit.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | try  {  // Statements to be monitored for exceptions  }  //You can't keep statements here  catch(Exception ex)  {  //Cathcing the exceptions here  }  //You can't keep statements here  finally  {  // This block is always executed  } |

**5) Can we write only try block without catch and finally blocks?**

No, It shows compilation error. The try block must be followed by either catch or finally block. You can remove either catch block or finally block but not both.

**6) There are three statements in a try block – statement1, statement2 and statement3. After that there is a catch block to catch the exceptions occurred in the try block. Assume that exception has occurred in statement2. Does statement3 get executed or not?**

No. Once a try block throws an exception, remaining statements will not be executed. control comes directly to catch block.

**7) What is unreachable catch block error?**

When you are keeping multiple catch blocks, the order of catch blocks must be from most specific to most general ones. i.e sub classes of Exception must come first and super classes later. If you keep super classes first and sub classes later, compiler will show unreachable catch block error.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | public class ExceptionHandling  {  public static void main(String[] args)  {  try  {  int i = Integer.parseInt("abc");   //This statement throws NumberFormatException  }  catch(Exception ex)  {  System.out.println("This block handles all exception types");  }  catch(NumberFormatException ex)  {  //Compile time error  //This block becomes unreachable as  //exception is already caught by above catch block  }  }  } |

**8) Explain the hierarchy of exceptions in java?**

Click [here](https://javaconceptoftheday.com/hierarchy-exceptions-java/) to see the hierarchy of exceptions in java.

**9) 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.

**10) What is OutOfMemoryError in java?**

OutOfMemoryError is the sub class of java.lang.Error which occurs when JVM runs out of memory.

**11) what are checked and unchecked exceptions in java?**

Checked exceptions are the exceptions which are known to compiler. These exceptions are checked at compile time only. Hence the name checked exceptions. These exceptions are also called compile time exceptions. Because, these exceptions will be known during compile time.

Unchecked exceptions are those exceptions which are not at all known to compiler. These exceptions occur only at run time. These exceptions are also called as run time exceptions. All sub classes of java.lang.RunTimeException and java.lang.Error are unchecked exceptions.

Click [here](https://javaconceptoftheday.com/checked-unchecked-exceptions-java/) to see more about checked and unchecked exceptions.

**12) What is the difference between ClassNotFoundException and NoClassDefFoundError in java?**

Click [here](https://javaconceptoftheday.com/classnotfoundexception-vs-noclassdeffounderror-in-java/) to see the differences between ClassNotFoundException and NoClassDefFoundError in java.

**13) Can we keep the statements after finally block If the control is returning from the finally block itself?**

No, it gives unreachable code error. Because, control is returning from the finally block itself. Compiler will not see the statements after it. That’s why it shows unreachable code error.

**14) Does finally block get executed If either try or catch blocks are returning the control?**

Yes, finally block will be always executed no matter whether try or catch blocks are returning the control or not.

**15) Can we throw an exception manually? If yes, how?**

Yes, we can throw an exception manually using throw keyword. Syntax for throwing an exception manually is

**throw InstanceOfThrowableType;**

Below example shows how to use throw keyword to throw an exception manually.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | try  {  NumberFormatException ex = new NumberFormatException();    //Creating an object to NumberFormatException explicitly  throw ex;        //throwing NumberFormatException object explicitly using throw keyword  }  catch(NumberFormatException ex)  {  System.out.println("explicitly thrown NumberFormatException object will be caught here");  } |

Click [here](https://javaconceptoftheday.com/throwing-rethrowing-exception-java/) to see more about throw keyword.

**16) What is Re-throwing an exception in java?**

Exceptions raised in the try block are handled in the catch block. If it is unable to handle that exception, it can re-throw that exception using throw keyword. It is called re-throwing an exception.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | try  {  String s = null;  System.out.println(s.length());   //This statement throws NullPointerException  }  catch(NullPointerException ex)  {  System.out.println("NullPointerException is caught here");  throw ex;     //Re-throwing NullPointerException  } |

**17) What is the use of throws keyword in java?**

Click [here](https://javaconceptoftheday.com/throws-keyword-java/) to see the uses of throws keyword in java.

**18) Why it is always recommended that clean up operations like closing the DB resources to keep inside a finally block?**

Because finally block is always executed whether exceptions are raised in the try block or not and raised exceptions are caught in the catch block or not. By keeping the clean up operations in finally block, you will ensure that those operations will be always executed irrespective of whether exception is occurred or not.

**19) What is the difference between final, finally and finalize in java?**

Click [here](https://javaconceptoftheday.com/difference-between-final-finally-and-finalize-in-java/) to see the differences between final, finally and finalize in java.

**20) How do you create customized exceptions in java?**

Click [here](https://javaconceptoftheday.com/user-defined-exceptions-java/) to see about customized exceptions in java.

**21) What is ClassCastException in java?**

ClassCastException is a RunTimeException which occurs when JVM unable to cast an object of one type to another type.

**22) What is the difference between throw, throws and throwable in java?**

Click [here](https://javaconceptoftheday.com/difference-between-throw-throws-and-throwable-in-java/) to see the differences between throw, throws and throwable in java.

**23) What is StackOverflowError in java?**

StackOverflowError is an error which is thrown by the JVM when stack overflows.

**24) Can we override a super class method which is throwing an unchecked exception with checked exception in the sub class?**

No. If a super class method is throwing an unchecked exception, then it can be overridden in the sub class with same exception or any other unchecked exceptions but can not be overridden with checked exceptions.

**25) What are chained exceptions in java?**

Click [here](https://javaconceptoftheday.com/chained-exceptions-java/) to see about chained exceptions in java.

**26) Which class is the super class for all types of errors and exceptions in java?**

java.lang.Throwable is the super class for all types of errors and exceptions in java.

**27) What are the legal combinations of try, catch and finally blocks?**

1)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | try  {  //try block  }  catch(Exception ex)  {  //catch block  } |

2)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | try  {  //try block  }  finally  {  //finally block  } |

3)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | try  {  //try block  }  catch(Exception ex)  {  //catch block  }  finally  {  //finally block  } |

**28) What is the use of printStackTrace() method?**

printStackTrace() method is used to print the detailed information about the exception occurred.

**29) Give some examples to checked exceptions?**

ClassNotFoundException, SQLException, IOException

**30) Give some examples to unchecked exceptions?**

NullPointerException, ArrayIndexOutOfBoundsException, NumberFormatException

**Checked Exceptions**

Checked exceptions are known to compiler i.e they are the exceptions that are checked at compile time. Checked exceptions are also called compile time exceptions, because they can be known during compile time.

**Unchecked Exceptions**

Unchecked exceptions are not known to compiler.  They are the exceptions that are not checked at compile time, because they occur only at run time.That’s why these exceptions are also called run time exceptions.

Now, come to hierarchy of exceptions in java.

**java.lang.Throwable :**

java.lang.Throwable is the super class of all errors and exceptions in java. Throwable class extends java.lang.Object class. The only argument of catch block must be it’s type or it’s sub class type. You can check the documentation of Throwable class [here](http://docs.oracle.com/javase/7/docs/api/java/lang/Throwable.html). It has two sub classes.

1)java.lang.Error

2)java.lang.Exception

**1) java.lang.Error :**

java.lang.Error is the super class for all types of errors in java. You can follow the documentation of Error class [here](http://docs.oracle.com/javase/7/docs/api/java/lang/Error.html). Some of the common errors are,

* [java.lang.VirtualMachineError](http://docs.oracle.com/javase/7/docs/api/java/lang/VirtualMachineError.html) : The most common virtualMachineErrors are [StackOverFlowError](http://docs.oracle.com/javase/7/docs/api/java/lang/StackOverflowError.html" \t "_blank) and [OutOfMemoryError.](http://docs.oracle.com/javase/7/docs/api/java/lang/OutOfMemoryError.html" \t "_blank)
* [java.lang.AssertionError](http://docs.oracle.com/javase/7/docs/api/java/lang/AssertionError.html)
* [java.lang.LinkageError](http://docs.oracle.com/javase/7/docs/api/java/lang/LinkageError.html)  : The common LinkageError are [NoClassDefFoundError](http://docs.oracle.com/javase/7/docs/api/java/lang/NoClassDefFoundError.html" \t "_blank) and subclasses of [IncompatibleClassChangeError](http://docs.oracle.com/javase/7/docs/api/java/lang/IncompatibleClassChangeError.html" \t "_blank). The most frequent IncompatibleClassChangeErrors are NoSuchMethodError, NoSuchFieldError, AbstractMethodError, IllegalAccessError and InstantiationError.

All sub classes of Error class are unchecked type of exceptions. i.e They occur during run time only.

**2) java.lang.Exception :**

java.lang.Exception is the super class for all types of Exceptions in java. You can follow the documentation of Exception class [here](http://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html). All sub classes of Exception class except sub classes of RunTimeException are checked type of exceptions. Some of the common sub classes of Exception are,

* [java.lang.RunTimeException](http://docs.oracle.com/javase/7/docs/api/java/lang/RuntimeException.html)

All sub classes of RunTimeException are unchecked type of exceptions. i.e They occur during run time only. Some common RunTimeException are [ArithmeticException](http://docs.oracle.com/javase/7/docs/api/java/lang/ArithmeticException.html), [NumberFormatException](http://docs.oracle.com/javase/7/docs/api/java/lang/NumberFormatException.html), [NullPointerException](http://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html), [ArrayIndexOutOfBoundsException](http://docs.oracle.com/javase/7/docs/api/java/lang/ArrayIndexOutOfBoundsException.html) and [ClassCastException](http://docs.oracle.com/javase/7/docs/api/java/lang/ClassCastException.html).

* [java.lang.InterruptedException](http://docs.oracle.com/javase/7/docs/api/java/lang/InterruptedException.html)
* [java.lang.IOException](http://docs.oracle.com/javase/7/docs/api/java/io/IOException.html)
* [java.lang.SQLException](http://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html)
* [java.lang.ParseException](http://docs.oracle.com/javase/7/docs/api/java/text/ParseException.html)

All above classes are placed in java.lang package. Click on class to follow the documentation of that class