**Finally Block**

* When we write System.exit() in try block

Difference between Checked and un checked exceptions

|  |  |
| --- | --- |
| **Checked** | **Un- Checked** |
| The compiler checks a checked exception  These types of exceptions can be handled at the time of compilation.  Compiler will force the programmer to surround the statements with try and catch block | The compiler does not check these types of exceptions  These types of exceptions cannot be a catch or handle at the time of compilation, because they get generated by the mistakes in the program.  Compiler will not force the programmer to surround the statements with try and catch block |

**Examples of checked Exception**

* File Not Found Exception
* Class Not Found Exception
* No Such Method Exception
* No Such Field Exception

**Finally block without try block**

No we cannot use finally without try

**Difference between Final Finally and Finalise**

|  |  |  |
| --- | --- | --- |
| **Final** | **Finally** | **Finalise** |
| Final is the keyword and access modifier  Final keyword is used with the classes, methods and variables  Once declared, final variable becomes constant and cannot be modified. (2) final method cannot be overridden by sub class. (3) final class cannot be inherited.  Final method is executed only when we call it. | finally is the block in Java Exception Handling  Finally block is always related to the try and catch block in exception handling.  finally block runs the important code even if exception occurs or not. (2) finally block cleans up all the resources used in try block  Finally block is executed as soon as the try-catch block is executed. | finalize is the method in Java which is used to perform clean up processing just before object is garbage collected.  finalize() method is used with the objects.  finalize method performs the cleaning activities with respect to the object before its destruction.  finalize method is executed just before the object is destroyed. |

**Difference Between Exception and error**

|  |  |
| --- | --- |
| **Exception** | **Error** |
| An exception is recoverable.  Errors can occur at compile time as well as run time. Compile Time: eg Syntax Error  As the error is detected the program will terminated abnormally.  Errors are classified as unchecked type.  In Java, errors are defined "java.lang.Error" package. | An error is irrecoverable.  As an exception is detected, it is thrown and caught by the "throw" and "catch" keywords correspondingly.  Exceptions are classified as checked or unchecked type.  In Java, an exceptions are defined in"java.lang.Exception". |

**Custom Exceptions**

* Programmer can also create his own exceptions which are called user-defined exception or custom exceptions
* In java in built exceptions are not able to describe a certain situations

**Steps for creating**

* Class should inherit RuntimeException (or) Exception
* Throw keyword is used the object of user defined exception class

**Re-Throwing An Exception**

* We can re-thrown exception from catch block to another class
* It is called exception propogations
* Throws Keyword is used

**Can we Write Statements Between Try Catch Finally**

* No, we cannot write any statements in between try, catch and finally blocks and these blocks form one unit**.**

**ClassNotFound Exceptions**

* Thrown when an application tries to load in a class through its string name using:
* The forName() method in class Class.
* The findSystemClass() method in class ClassLoader .
* The loadClass() method in class ClassLoader.
* but no definition for the class with the specified name could be found**.**