* ***what is difference between exception and error?***
* *Exception can be handled through program and error cannot be handled through program.*
* *Exceptions are related to the application and an Error is related to the environment in which the application is running.*
* *An Error can't be recovered as it is fatal in nature, that is different in the case of an Exception that may not be fatal in all cases.*
* ***how can we handle exceptions in java***

*we can handle exceptions using 5keywords …those are try, catch, throw , throws, finally*

* *try- try is used to specify the block of code that can throw an exception.try block must be followed by catch or finally.*
* *Catch- it is used to process the exceptions thrown by try block.*
* *Throw- this keyword is used to create new exception instance.*
* *Throws-throws keyword is used to declare what kind of exceptions can we expect when executing the method.*
* *Finally-it can guarantee that a section of code will be executed, even if an exception is thrown*
* ***why do we need exception handling***

*Exception is nothing but converting system error generated message into user friendly error message.* *These system generated messages are not understandable by user so need to convert it into user friendly error message. So we need exception handling feature to convert the error message.*

* ***what is exception hierarchy***

*In Java “an event that occurs during the execution of a program that disrupts the normal flow of instructions” is called an exception. This is generally an unexpected or unwanted event which can occur either at compile-time or run-time in application code. Java exceptions can be of several types and all exception types are organized in a fundamental hierarchy.*

***Throwable***

***Unchecked***

***Checked***

***Exceptions***

***Errors***

***ArthimeticException***

***NullPonter***

***ArrayIndexOutOfBound***

***SQLException***

***IOException***

***FilenotfoundException***

* ***how can we create user defined runtime and compile time exceptions***

*User Defined Exception or custom exception is creating your own exception class and throws that exception using ‘throw’ keyword. This can be done by extending the class Exception.*

*create a****custom compile time exception****all you have to do is to extend Runtime****Exception****class.*

* ***what is try with resources***

*The try-with-resources statement is a try statement that declares one or more resources. A*resource*is an object that must be closed after the program is finished with it. The try-with-resources statement ensures that each resource is closed at the end of the statement. Any object that implements java.lang.AutoCloseable, which includes all objects which implement java.io.Closeable, can be used as a resource.*

* ***can we have try, try and finally without catch block***

*Yes, It is possible to have a try block without a catch block by using a final block*. *Because finally block will always execute even there is an exception in try block.*

* ***explain the chart and explain the throwable error exception relation***

*All errors and exception classes are the subclasses of java.lang.Throwable which further extends the java.lang.Object class.*

* ***what is getMessage and printstacktrace***

*This printstacktrace method will display the name of the exception and nature of the message and line number where an exception has occurred*.

*This getMessage method will display the only exception message*.

* ***how to write proper exception handling in java***

*best practices to handle exceptions in java*

### *Use a Finally Block*

*In contrast to the last few lines of your try block, the finally block gets always executed. That happens either after the successful execution of the try block or after you handled an exception in a catch block. Due to this, you can be sure that you clean up all the opened resources.*

## *Prefer Specific Exceptions*

## *The exception’s message gets read by everyone who has to understand what had happened when the exception was reported in the log file or your monitoring tool.*

## *Don’t Catch Throwable*

*Throwable is the superclass of all exceptions and errors. You can use it in a catch clause, but you should never do it! If you use*Throwable*in a catch clause, it will not only catch all exceptions; it will also catch all errors.*