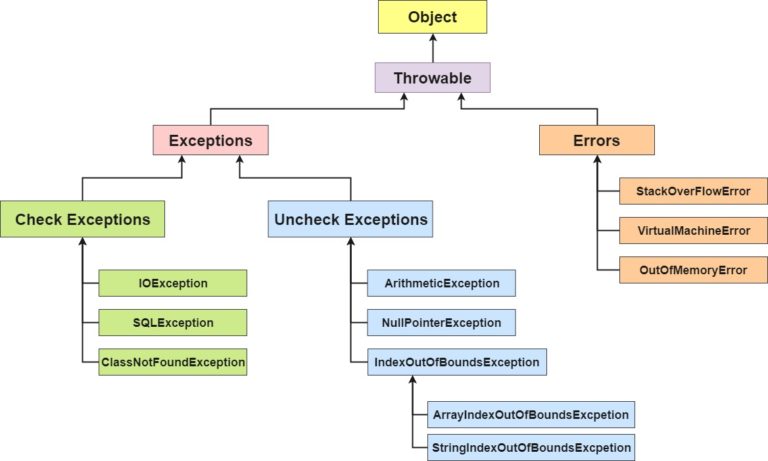
* Introduction :
* An unexpected unwanted event that disturbs normal flow of the program is called Exception
* Why we need to handle Exception
* It is highly recommended to handle exception the main objective of exception handling is graceful termination of the program
* Exception handling does not mean repairing an exception we have to provide alternative way to continue rest of the program normally it’s the concept of exception handling
* Our requirement is to read data from remote file at runtime if the file is not available our program should not be abnormally we have to provide some local file to continue to rest of the program normally this way of defining it’s exception handling
* Inside a method any exception occurs the method in which rised is responsible to create an exception object by including the following information
* Name of exception
* Description of exception
* Location at which exception occurs(stack trace)
* After creating an exception object method handover the object to the jvm
* Jvm will check any exception handling code or note if the method does not contain exception handling code then jvm terminates the method abnormally and removes corresponding entries from the stack.
* Exception in thread xxx ,name of the exception, description and stack trace
* In program at least terminates abnormally then the program termination is abnormal termination
* If all methods terminated normally then only program termination is normal termination



* Exception hierarchy
* Thwroable class has access root for java exception hierarchy
* Thwroable class defines two child classes
  + Exception
    - Most of the times exceptions are caused by our program and these are recoverable
    - Ex :if our program need remote file and at runtime if remote file is not available then we can use local file
  + Error
    - Most of the times error are not caused by our program and these are due to lack of system resources
    - Errors are not recoverable
    - If OutOfMemory occurs being a programmer we can’t do anything and the program will be terminated abnormally
* RuntimeException
  + Aritmaticexception
  + NullPointerException
  + ClassCastException
  + IndexOutOfBoundException
  + ArrayIndexOutOfBound
  + StringIndexOutOf
* ServletException
* SqlException
* StackOverFlowError
* OutOfmemoryerror
* Exception (checked and unchecked)
* Checked : checked exception which has to be handle at compile time PrintWriter v= new PrintWriter()
* Fully checked vs. partially checked
* A checked exception is said to be fully check if and only if all it’s child classes also check ex :IOException,Intruppted exception
* A checked exception is said to be partialy check if and only if all it’s some child classes are not checked

Note: when we print any object reference internally toString method get called.

* PrintStackTrace()
* Print all things(complete info), name of exception, description, stack trace
* toString() or e(object reference)
* name of exception, description
* Getmessage()
* Description only
* Internally default exception will use printstacktrace method to print exception info to the console
* Try with multiple catch block:
* Order of catch block should be correct (Child then parent)
* Finally :
* Which is block to maintain cleanup code(ex: database connection) which is use to clean try block resources
* Finalize is a method always invoked by garbage collector just before destroying an object to perform cleanup activities. Once finalize complete immediately garbage collector destroys that object.
* Throw
* Sometimes we can create explicitly we can handover to the jvm manually for this we have to use throw keyword Ex: throw new ArthmeticException(“/ by zero”);
* Hence the men objective is to handover over created exception object manually
* Best use of throw keyword is for user-defined exceptions are customized exception
* After throw statement we are not allowed write any statement directly otherwise we will get compile time error saying unreachable statement.
* We can use throw keyword only for Thwroable type if we are trying with normal java object we’ll get compile time error saying incompatible types
* Throws
* To delegate the exception handling responsibilities to the caller
* Throws keyword required only for checked exceptions and usage of unchecked exception there is no use or there is no impact
* Throws keyword required to convince only the compiler
* Usage of throws keyword does not prevent abnormal termination program
* .
* Try : to handle risky code
* Catch: to handle exception handling code
* Finally: to maintain cleanup code
* Throw: to handover our created exception to the jvm manually
* Throws: to delegate responsibilities to the caller method
* Various possible compile time error
* Unreachable statement
* Exception already been caught
* Incompatible types found test required: java.lang.throwable
* Try without catch or finally
* Catch without try
* Customize or user define exception
* Super(): to make description available to default exception handler
* ArrayIndexOutofBoundEsception child of RE unchecked
* NullPointerException child of RE unchecked
* ClassCastExceptin child of RE unchecked type cast parent to child
* IlligalArgumentException child of RE
* NumberFormatException child of illigalArgumentException child of RuntimeException
* StackOverFlow child of Error unchecked
* NoClassdefFoundError: whenever jvm unable to find required .class file