

Exception

Types of Errors:

- Compile time errors(Syntax errors)
- Run time errors(wrong logic)

Exception

- If the exception object is not caught and handled properly, the interpreter will display an error message.
- If we want the program to continue with the execution of the remaining code, then we should try to catch the exception object thrown by the error condition and then display an appropriate message for taking corrective actions, called exception handling.

CONTENT OF EXCEPTION HANDLING CO

f exception handling are throwing an exception and catching it. This

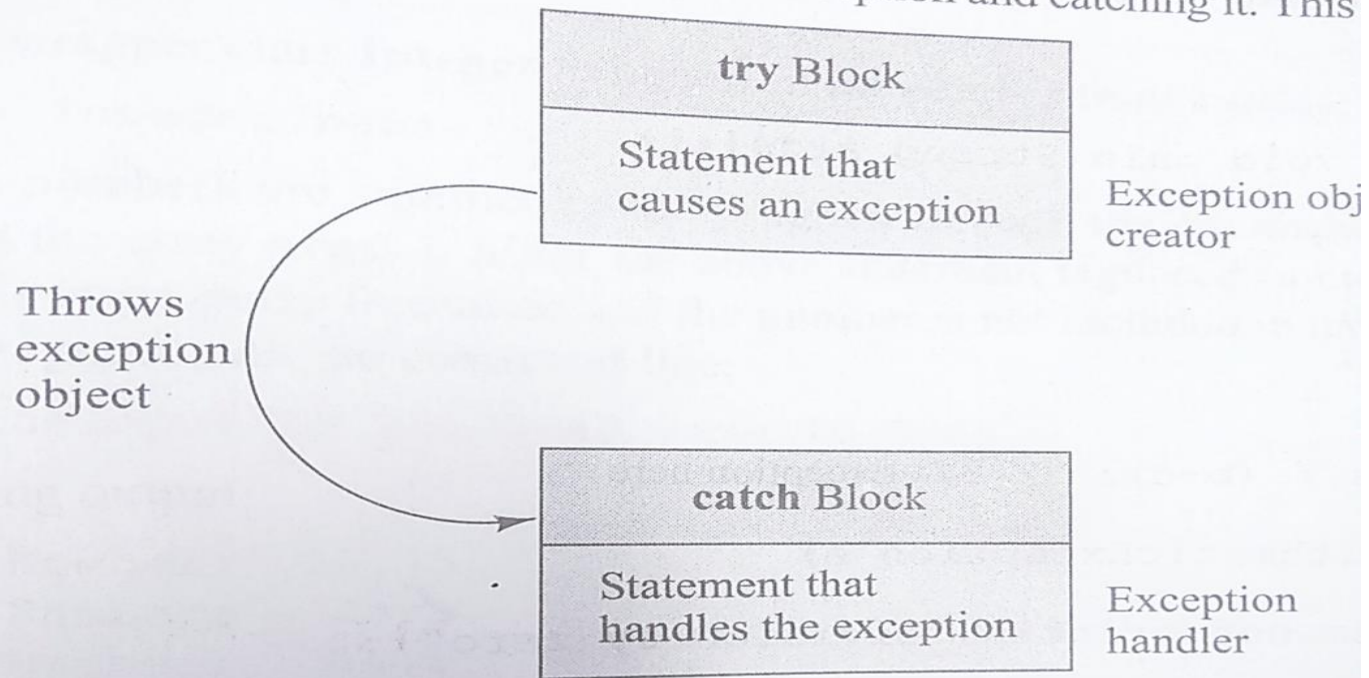


Fig. 13.1 *Exception handling mechanism*

Exception

- Exception handling:
 - Find the problem(HIT)
 - Inform that an error has occurred(Throw the exception)
 - Receive the error information(Catch the exception)
 - Take corrective actions(Handle the exception)

Exception

- Java Exceptions:
 - ArithmeticException
 - ArrayIndexOutOfBoundsException
 - ArrayStoreException
 - FileNotFoundException
 - IOException
 - NullPointerException
 - NumberFormatException
 - OutOfMemoryException
 - StringIndexOutOfBoundsException

Exception

- Two types of Exceptions:
 - Checked exceptions: Exceptions that are checked at compile time.

Checked exceptions are extended from the **java.lang.Exception**

- Unchecked Exceptions: These exceptions are not essentially handled in the program code JVM handles such exceptions.

Unchecked exceptions are extended from the **java.lang.RuntimeException**

Exception

.....

try

{

statement;

}

catch(Exception-type e)

{

statement;

}

.....

Multiple catch statements

.....

Try

{

statement;

}

catch(Exception-type1 e)

{

statement;

}

catch(Exception-type2 e)

{

statement;

}

catch(Exception-type3 e)

{

statement;

}

Finally statements

.....

Try

{

statement;

}

catch(Exception-type1 e)

{

statement;

}

catch(Exception-type2 e)

{

statement;

}

finally

{

statement;

}

throws keyword

- throws is a keyword in Java which is used in the signature of method to indicate that this method might throw one of the listed type exceptions.
- The caller to these methods has to handle the exception using a try-catch block.
- We can use throws keyword to delegate the responsibility of exception handling to the caller (It may be a method or JVM) then caller method is responsible to handle that exception.

throws keyword

- throws keyword is required only to convince compiler and usage of throws keyword does not prevent abnormal termination of program.
- By the help of throws keyword we can provide information to the caller of the method about the exception.

Throwing our own exception

- Create a class by extending Exception class and override toString method with the appropriate message.
- Throw an object using new (Throwable's subclass)
- Catch that object and display message

Throwing our own exception

- Write a program to accept number from the user and if the number contains digit 9 then it should throw an error message “Error: Number contains digit 9”.

Throwing our own exception

- Write a program which accepts even numbers from the user. If it is an even number then it displays “number entered successfully”, if not then it throws the exception of user defined class “InvalidNumberException”. The class should contain appropriate toString method to describe object.