**Exception Handling**

What is an Exception?

An exception is a situation which occurs in runtime leads to abnormal termination of execution of the program.

Here we have two types of exceptions----

i)Checked Exception.

ii) Unchecked Exceptions

**CHECKED EXCEPTIONS: The exceptions which checked by the compiler and which occurs at compile time is said to be Checked Exceptions.**

**UNCHECKED EXCEPTIONS: The exception which occurs at runtime environment and is not checked by the compiler is said to be unchecked Exceptions.**

**Hierarchy of Exception Classes**

**OBJECT**

**Exception Error**

* **Checked Exceptions/ Virtual Machine Error**

**Assertion Error**

**Compile Time Exceptions**

1. **IO Exceptions.**
2. **SQL Exceptions.**

* **Runtime Exceptions/Unchecked**

**Exceptions.**

1. **ArithmeticException**
2. **NullPOinterException**
3. **NumberFormatException.**
4. **ArrayIndexOutOfBoundException......**

**How to handle Exceptions?**

We can handle exceptions through the ways :

1. **Using TRY and CATCH.**
2. **Throw.**
3. **Throws.**
4. **Finally**

**TRY and CATCH:**

Suppose we have an Exception in the code at a particular line we have to keep that line in try block and immediately catch block initiated which helps in handling exception.

Suppose there is an another line of code after exception line then exception handled through catch block and remaining code has been closed.

Eg: class A{

Public Static void main(String [] args)

{

Try

{

System.out.println(10/0);

}

Catch (Exception e)

{

System.out.println(10/5);

}

}

}

**FINALLY:**

Here the finally block is executed after the exception is handled over by complier. And it is executed for sure at the end.

Eg: Class A

{

Public static void main(String[] args)

{

Try

{

System.out.println(10/0);

}

Catch(Exception e)

{

System.out.println(10/5);

}

}

}

**Throw:**

The throw is used when there is multiple exceptions in code and to avoid them we use throw and handle them by using try and catch statements.

Syntax: class A {

Public void add() throw arithmeticException

{

System.out.println(10/0);

}

Public static void main(String[]args)

{

Try{

A a = new A();

a.add();

}

Catch (Exception e)

{

System.err.println(e);

}

}

Here we used system.err.println to print what is the exception caught and which line the exception has been caught .We can use to print exceptions in three ways.

1. **System.out.println(e);**
2. **System.err.println(e);**
3. **E.printStackTrace();**

We have many ways to handle exceptions but the best thing is to handle with try and catch block only.