**Exception:** When abnormal situation occurs, JVM will terminate the program by throwing exception class object.

**Exception Handling:** To skip instruction with exception and execute remaining part of the program correctly, to stop in appropriate termination of the program.

For exception handling we can use:

try{

risky code;

}

catch (Exception class name reference){

msg related to exception

}

**Syntax Error:**

If we miss any ( ; ) or ( { ) in the code it will give syntax error.

**Logical error:**

A=10,b=5

Id(a>b)

System.out.println(“b is max”);

**Exception is classified into two types: syntax error.**

1. Checked exception: Unknown to the compiler.
2. Unchecked exception: Known to the compiler

Get message(); = only gets message

toString(); = method to convert object into message/prints class name and message

printStackTrace(); = trace the line where exception occurs

//catch (Exception x){

//x.printStackTrace();

//}

Try Catch:

1. Single try catch
2. Multiple try single catch
3. Single try multiple catch
4. Nested try catch

**Finally:** It is a block which is used to deallocate the memory or close the objects. It can be written only with try catch.

**Throw:** It is used to generate user defined exception. It can only handle checked exceptions.

\*Throw and throws are different.

Throw is a keyword used in the method body to throw an exception.

Throws is used in method to declare the user-defined exception.

Throwable

🡪Exception >Runtime exception >Compile time exception

🡪 Error