⇒ Control flow in Try-catch

1)
```
try
{
    stmt 1;
    stmt 2;
    stmt 3;
} Catch (Ex. e)
{
    Stmt 4;
}
Stmt 5;
```

Case 1:- If there is no exception :-

1, 2, 3, 5 (Normal Termination)

Case 2:- If there is Exception at stmt 2 & matching catch block is present.

1, 4, 5 (Normal Termination.

Case 3:- if an exception raised at stmt 2 and corresponding catch block not matched:-

1, (Abnormal Termination.

Case 4:- if an Exception raised at stmt 4, or stmt 5

⇒ Abnormal Termination.

~~Case 5:-~~ NOTE :-

① Within the try-block if anywhere Exception raised, then rest of the try block won't be executed even though we handled that exception. Hence, within the try-block, we have to take only the risky code and length of try block should be as less as possible.

(ii) In addition to try block, there may be a chance of raising an exception inside catch and finally blocks.

(iii) If any statement which is raising an exception and it is not part of try block then it is always abnormal termination.

Ep-39, 33:01

⇒ Methods to print Exception Information:→

→ Throwable class defines the following methods to print exception information.

Method	Printable Format
(i) printStackTrace()	Name of Exception : Description Stack Trace
(ii) toString()	Name of Exception : Description
(iii) getMessage()	Description

Ex:-

```
class Test {
    p s v main(String[] args)
    {
        try {
            S.O.p(10/0);
        } catch (ArithmeticException e)
        {
            e.printStackTrace();
            S.O.p(e); or S.O.p(e.toString());
            S.O.p(e.getMessage());
        }
    }
}
```

Java.lang.AE: / by zero
at Test.main

Java.lang.AE: / by zero

/ by zero