# Exception Handling

→ What is exception?

unexpected event disturbs normal flow of
program.

→ egt  FileNotFoundException

→ What is Purpose of exception handling?

graceful termination

egt  | Open db Connection
     | Read data ——→ SQLException,,
     ↓ Close db Connection

→ In what ~~any~~ way you are handling exception?

defining alternative way to continue rest of
the program normally is called exception
handling.

eg ① Read remote file       | ✗ Read remote file
       fileNotFoundException()  | if (RemoteFile not availble)
                                 | {
                                 | else Local File;
                                 | }

② try {
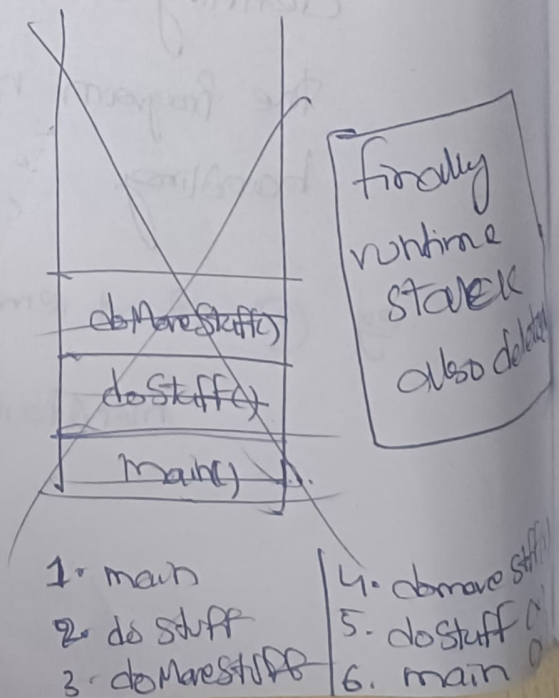
Read remote File

} catch ( File Not Found Exception e) {

use local ~~frame~~ file ;

}

---

## Runtime Stack mechanism

→ For every thread has ~~only~~ own stack



[JVM] Creates stacks

$t_1$

$m_1$ → Stackframe / Activation Record
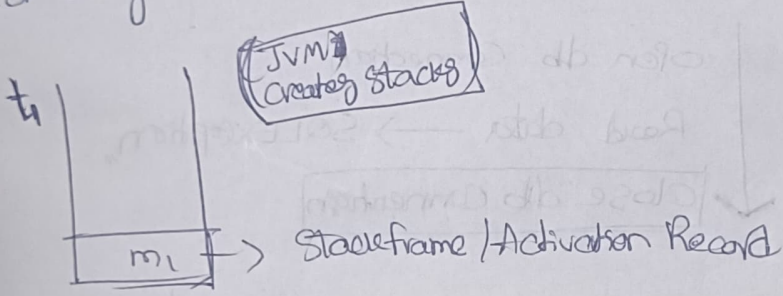
eg + class Test
{
    P S V m ( String [ ] args )
    {
       doStuff ( ) ;
    }

    P S v doStuff ( ) {

       doMoreStuff ( ) ;
    }

    P S v doMoreStuff ( ) {

       S. o. P ( Hi ) ;

    }

main thread

~~doMoreStuff( )~~

~~doStuff( )~~

main( )

finally
runtime
stack
also deleted

| | |
|---|---|
| 1. main | 4. domore Stuff |
| 2. do stuff | 5. do stuff ( ) |
| 3. doMoreStuff | 6. main ( ) |

# Default exception Handling in Java

```
Class Test
{
    P S V main (String [] args)
    {
        dostuff();
    }

    P S V doStuff ()
    {
        doMorestuff();
    }

    P S V doMorestuff ()
    {
        S.o.p (10 | 0);  ──→  The method in which
                               exception raised is
                               responsible to create
                               exception object with JVM
                                                    help
    }
}
```

exception object ──→ | Name: AExcp |
                     | Description: / by zero |  handover object to
                     | Stacktrace : |              JVM
                                  P

                exception raised
Q) JVM ask, method do you have handling code ?
                 X

A) is NO, Method is terminated abnormally
   without executing remaining code and remove
   method from stack. and control now goes to
   method which called the exception causing method
   ie, doStuff method in our example

if no handling code in caller method
that method also abnormally terminated
and entry is removed from stack

Finally control goes to main method

then if main method also didn't
handle exception its also terminated
abnormally.

Now Finally control comes to jvm so its
responsible to handle using "default
exception handling"

default exception handling print to console
in specific format and terminate,

Format of Printing Default exception

exception in Thread 'main' : AE : / by zero
   at Test. doMoreStuff()
   at test. doStuff()
   at test. main()