

# Exception Handling

→ What is exception?

unexpected event disturbs normal flow of program.

→ e.g. FileNotFoundException

→ What is Purpose of exception handling?

graceful termination

e.g.  
↓  
open db connection  
Read data → SQLException  
↓  
close db connection

→ In what ~~way~~ way you are handling exception?  
defining alternative way to continue rest of the program normally is called exception handling.

e.g. ① Read remote file  
FileNotFoundException  
|  
~~① Read remote file  
if (Remote file not available)  
{  
use Local file;  
}~~

2. try {

Read remote file

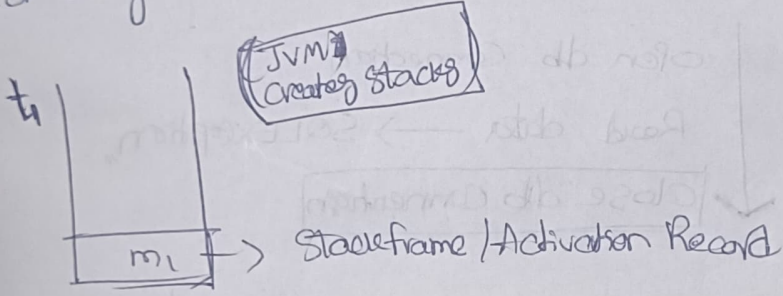
} catch (FileNotFoundException e) {

use local file;

}

### Runtime Stack mechanism

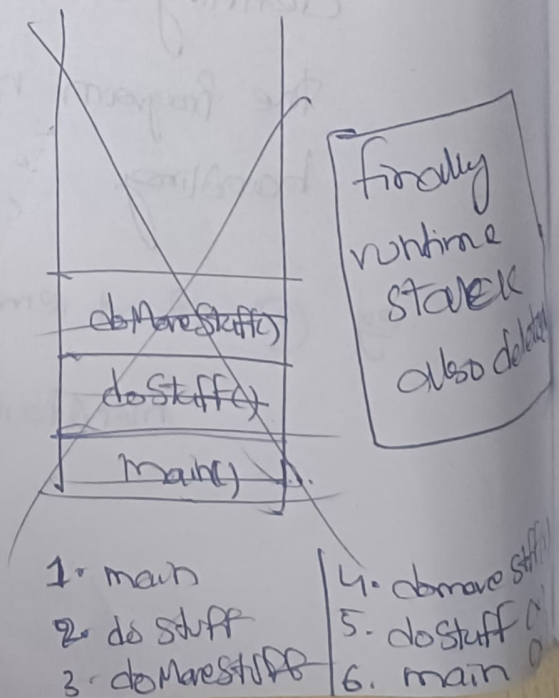
→ For every thread has ~~only~~ own stack



eg: Class Test

```
{  
    PSVM (String [] args)  
    {  
        doStuff();  
    }  
    PSV doStuff() {  
        doMoreStuff();  
    }  
    PSV doMoreStuff() {  
        S.O.P (Hi);  
    }  
}
```

main thread





# Default Exception Handling in Java

class Test

{

public main (String [] args)

{

doStuff();

}

public doStuff()

{

doMoreStuff();

}

public doMoreStuff()

{

S.o.p(10/0);

}

The method in which exception raised is responsible to create exception "object" with jvm help

exception object →

Name: AExcep  
Description: 1 by zero  
Stacktrace:

handover object to jvm

Q) JVM ask <sup>exception raised</sup> method do you have handling code?

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 i.e. 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 it's also terminated  
abnormally.

Now Finally control comes to JVM so it's  
responsible to handle using "default  
exception handling".

default exception handling prints the error  
in specific format and terminate

### Format of Printing Default Exception

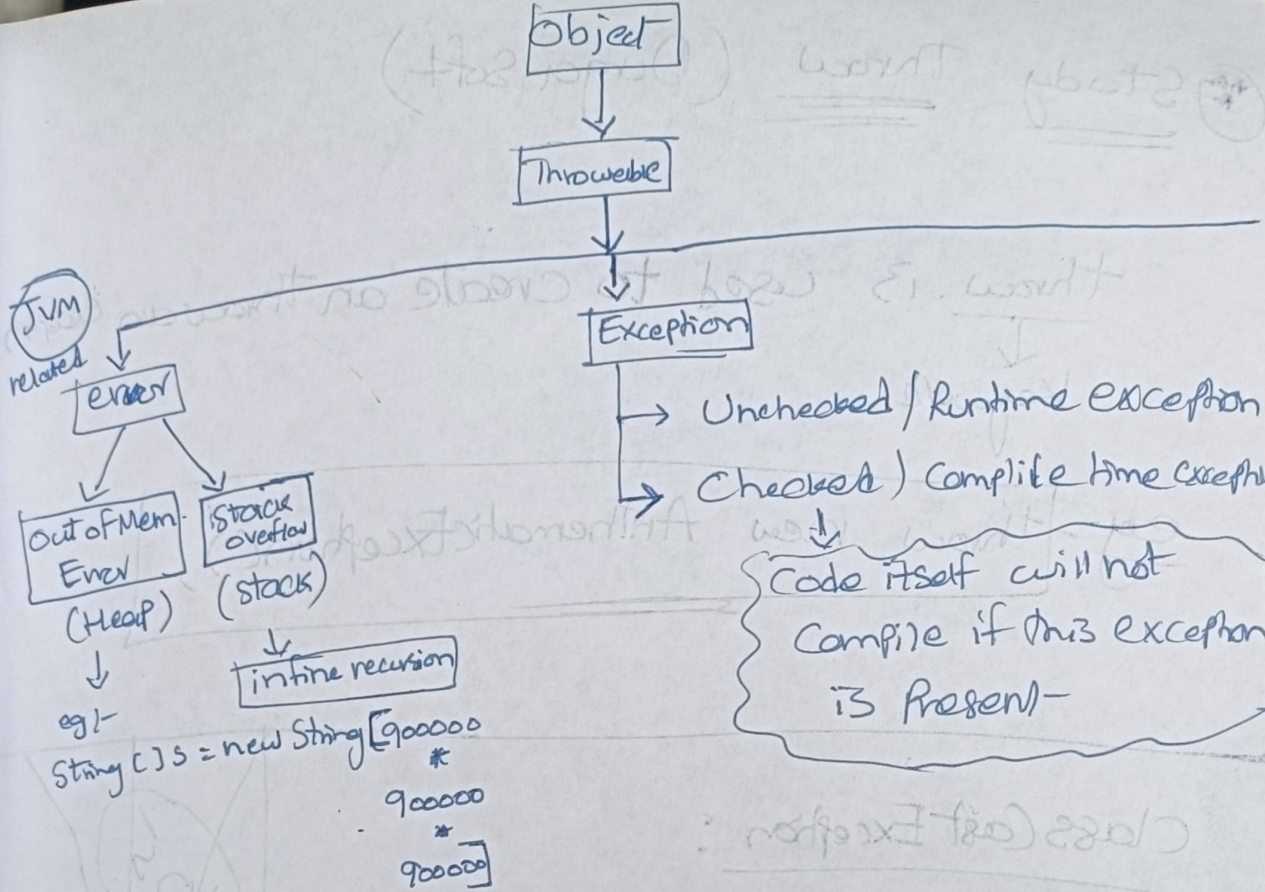
exception in Thread 'main': AE: / by zero

at test.demostuff();

at test.destuff();

at test.main();





## Runtime - Exception (unchecked)

- ClassCastException
- ArithmeticException
- IndexOutOfBoundsException
- NullPointerException
- IllegalArgumentException
  - ↳ NumberFormatException

occur during runtime  
Compiler is not forcing  
to handle them

RTE

## Compiletime - Exception

- ClassNotFoundException
- InterruptedException
- IOException
  - ↳ FileNotFoundException
  - ↳ EOFException
- SQLException

Compiler Verifies during  
Compilation and Code Compilation  
will fail

# ④ Study Throw (Durga Soft)

throw is used to create an thrown exception  
↓  
Keyword

eg. throw new ArithmeticException();

Class Cast Exception :

```
psvm(S-r args)
```

```
{
```

```
Object val = 10;
```

```
S.O.P((String)val)
```

```
}
```

Cannot "cast" String to Integer

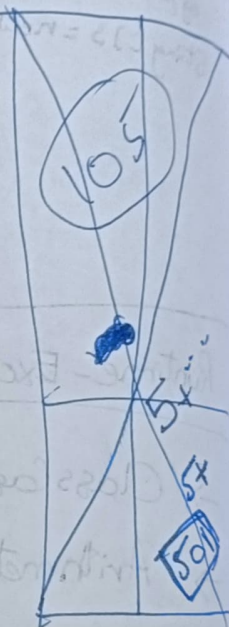
Arithmetic Exception

```
psvm(S-r args)
```

```
{
```

```
S.O.P(10/0);
```

```
}
```





## Index Out of Bounds (String/array):

```
Psum(arr S_[] args)
```

```
{  
    int[] a = new int[3];
```

```
    sop(a[3]);  
}
```

## Null Pointer Exception :-

```
Psum(S_[] args)
```

```
{
```

```
    String s = null;
```

```
    s.charAt(0); // Null Pointer Exception
```

```
}
```

## Illegal Argument Exception → (Number Format Excep)

```
Psum(S_[] args)
```

```
{
```

```
    System.out.println(Integer.parseInt("abc"));
```

```
}
```

## Compiletime Exception

→ Compiler checks & Compilation will fail if not handled properly

eg:- "ClassNotFoundException"

class Main

{

PS VM (String[] args)

{

mic);

}

PS VM mic)

{

throw new ClassNotFoundException();

}

}

Fix-1 for compile time exceptions

1. Throws : It tells that, this method  
↓  
keyword "MIGHT" throw this ~~error~~ exception  
(or might not) so caller  
you handle it appropriately
- Can be used  
any after  
method