

- error
- exception

exception is a unwanted or unexpected event which occurs during the program execution, which disturbs the normal flow of the execution.

① ②

unintentionally

Code

work

0_p

Acceptable

X

5 Risky Code →

Risky code

```

3
catch (*** e) {

```

alternative code to handle exception

3

8 4 2 1

0 1 0 0 0-1
0 0 0 0

00-
60-2000
15

```
public static void main(String[] args) {
    System.out.println("Enter a and b value.");
    Scanner sc = new Scanner(System.in);
    int a = sc.nextInt();
    int b = sc.nextInt();
    try {
        System.out.println(a/b);
    } catch (Exception e) {
        System.out.println("infinity..");
    }
    System.out.println("Main Completed...");
}
```

$$a=10, L=0$$

Exception Object

if (b == 0) {
 S.O.P ("Enter proper value");
}
S.O.P (a/b)

1 change

```
int fun() {
```

```
return 10
```

instance

```
int b = fun()
```

Notes:

Notes:
→ In the above program there is every possibility the exception may occur.

in the line SOP(a/b), when ever if the user enters the value $b = 0$

→ If the exception occurs, the program would terminate abnormally

→ To overcome this problem we need to handle the exceptions

→ The exception can be handled with the help of `{try, catch}` blocks

→ when there is an exception thrown, put that code/line like `so p(a/b)` inside try block

→ write the alternative solution in the catch block to handle the exception

→ the catch always follow the try block

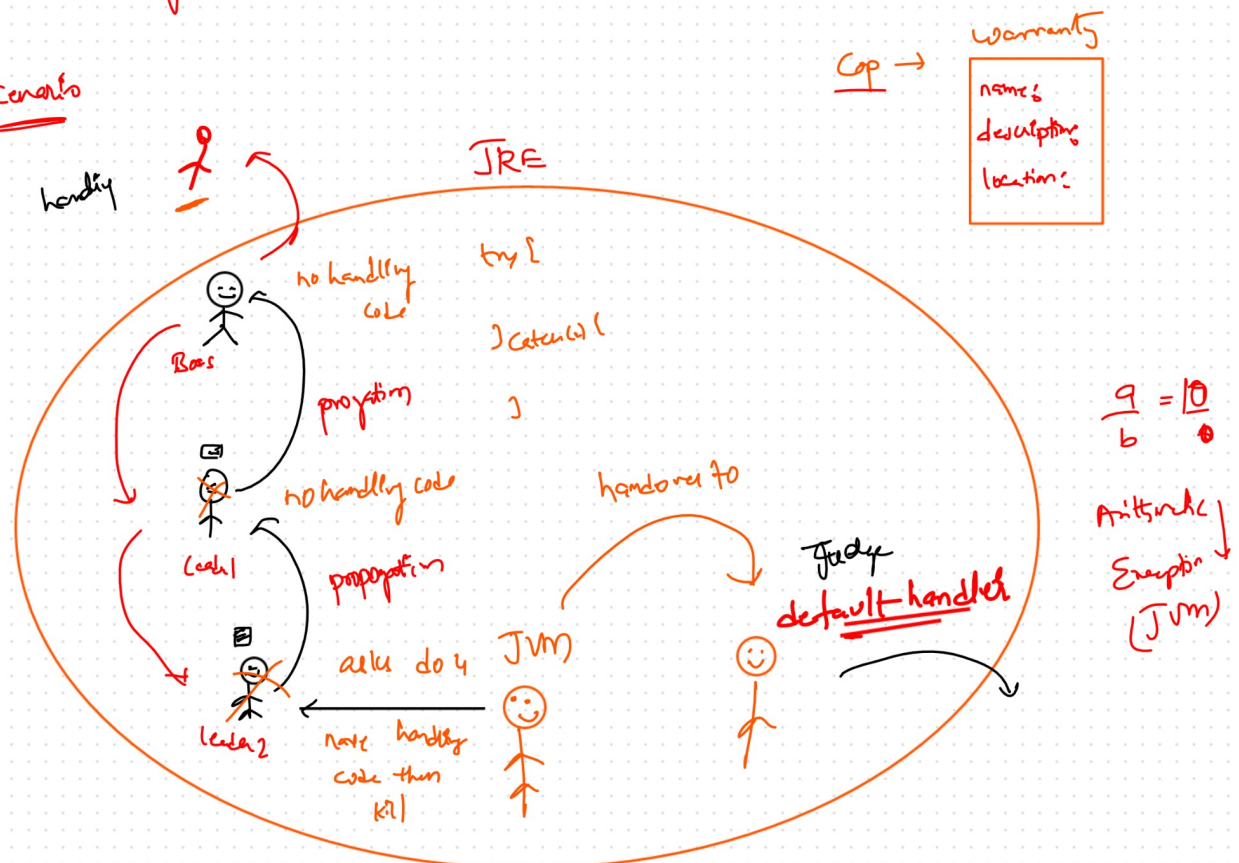
→ if any exception occurs inside the try block, the control would directly go to the catch block

→ After the execution of the catch block, the program flows normally

Outcome's

- (1) Gracefull termination [i.e. avoiding abnormal termination without seeing exceptions]
- (2) Providing awareness to the users (consumers) about the situation

Scenario



Runtime Stack

class Alpha {

private static void main(String args[]) {

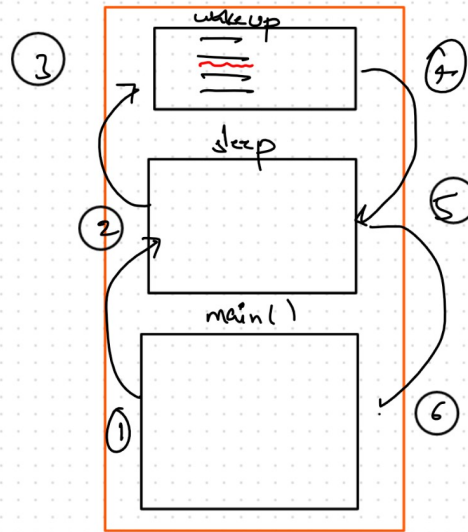
→ sleep();
}

public static void sleep() {

→ wakeup();
}

public static void wakeup() {

→ sop(100);
→ sop("Good Morning");
}



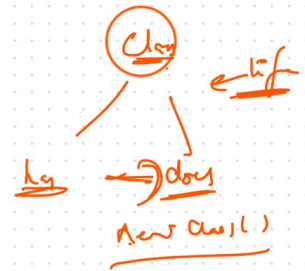
✓ try catch
✗

✗ Jvm ✓

Default Exception

handler

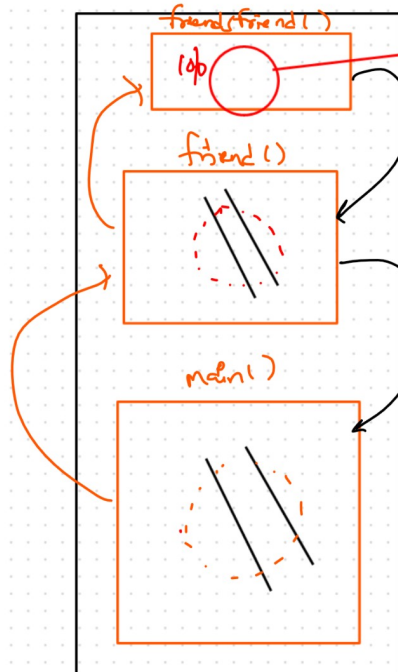
RTS



```
public class ExceptionBackPropagation {
    static void friendsFriend() {
        System.out.println(10 / 0);
    }
    static void friend() {
        friendsFriend();
    }
    public static void main(String[] args) {
        friend();
    }
}
```



default Exception handler



RTS

name:
description: The exception
StackTrace
Object

Jvm