Exception Handling

Exception

An Exception is a unwanted unexpected event that disrupts the normal flow of a program E.g -power failure,net disconnectivity.

Daily life Example -Suppose you are downloading something or watching a video online suddenly internet gets disconnected this is a exception.

Technical example -

- suppose you have written a program which reads a file on you system and print exactly on the screen.
    - After some time some one has deleted this File
    - Now when run your java file it will try to find on your system
    - Since the file is not there it will give FileNotFoundException

- You are building a calculator program
    - You have a method divide which divides two numbers a/b and returns it
    - Suppose at a point of time b (divisor) becomes 0
    - It will throw AirthmaticException

Exception Handling

The purpose of exception handling is if something goes wrong it prevent abrupt program crashes and maintain graceful flow/termination of program.

Daily life example - suppose you a vacation and you booked a train ticket which is in waiting,so if the ticket does not get confirmed you will take a bus for the same place, Here what happen you handled a exception,you are still going instead of cancelling or terminating your program

Technical example-

Lets us take the previous two example s of exception

- Before printing the data we will check if the file is found then only we will print the data other wise we will run our program further normally
    - Since this finding file is prone to exception
    - We will handle this and have a alternative way ready in the beginning only

- Suppose if the divisor is 0,we will handle this and return some alternative value or a message that we cannot divide it by 0.

```
//….other logic
//…..


try{

//Read data from file(risky code)

}catch(FileNotFileException ex){

//print some default message on continue you program normally

}

//…..
//…..
```