LET'S THINK FOR A BIT ABOUT HOW EXCEPTIONS ARE HANDLED

(THIS EXAMPLE IS FROM JAVA, BUT PRETTY MUCH ALL LANGUAGES DO IT THIS WAY)

LET'S SAY THERE IS A METHOD, WHERE THINGS CAN GO WRONG

```
public static void methodOne(int randomNumber) throws
        IOException, NullPointerException {
    if(randomNumber == 1) {
        throw new IOException("IO Exception");
    } else if (randomNumber == 2) {
        throw new NullPointerException("Null Pointe
    }
}
```

LET'S SAY THERE IS A METHOD,
WHERE THINGS CAN GO WRONG

```
public static void methodOne(int randomNumber) throws
        IOException, NullPointerException {
    if(randomNumber == 1) {
        throw new IOException("IO Exception");
    } else if (randomNumber == 2) {
        throw new NullPointerException("Null Pointer Exception");
    }
}
```

IN FACT THINGS CAN GO WRONG
IN 2 SPECIFIC WAYS -

THE CODE INSIDE THE FUNCTION
COULD LEAD TO 2 TYPES OF
EXCEPTIONS BEING THROWN,
SO THE FUNCTION VERY
HELPFULLY CALLS THIS FACT OUT
IN ITS SIGNATURE

THEN - ANY CODE CALLING THIS
METHOD MUST EITHER "CATCH"
THESE EXCEPTIONS, OR MUST
ANNOUNCE THAT IT TOO MIGHT
THROW THE EXCEPTION

}

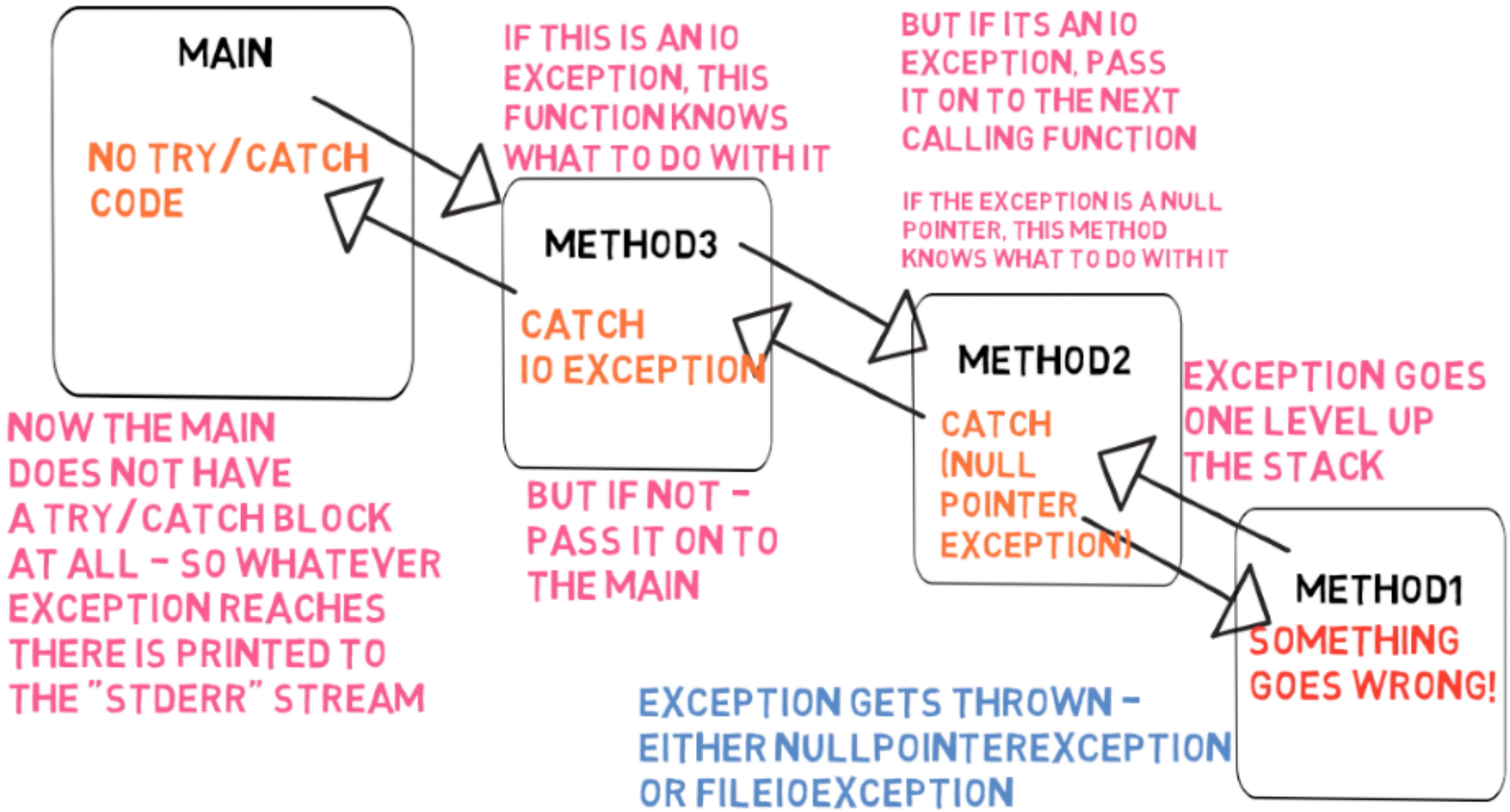
THIS METHOD KNOWS HOW TO HANDLE ONE OUT OF THE TWO TYPES OF EXCEPTIONS..

AND THIS METHOD IN TURN IS CALLED BY YET ANOTHER WHICH KNOWS WHAT TO DO WITH THE REMAINING TYPE OF EXCEPTION

```
public static void methodThree(int randomNumber) {
    try {
        methodTwo(randomNumber);
    } catch (IOException iox) {
        System.out.println("Caught the IO Exception inside methodThree");
    }
}
```

SINCE ALL OF THE ANNOUNCED TYPES OF EXCEPTIONS ARE NOW HANDLED, THE CODE THAT CALLS THIS METHOD NEED NOT DO ANYTHING AT ALL -

```
public static void main(String[] args) {
    int randomNumber = (int)Math.ceil(Math.random() * 10);
    methodThree(randomNumber);
}
```


MAIN
NO TRY/CATCH CODE
NOW THE MAIN DOES NOT HAVE A TRY/CATCH BLOCK AT ALL - SO WHATEVER EXCEPTION REACHES THERE IS PRINTED TO THE "STDERR" STREAM
IF THIS IS AN IO EXCEPTION, THIS FUNCTION KNOWS WHAT TO DO WITH IT
METHOD3
CATCH IO EXCEPTION
BUT IF NOT - PASS IT ON TO THE MAIN
BUT IF ITS AN IO EXCEPTION, PASS IT ON TO THE NEXT CALLING FUNCTION
IF THE EXCEPTION IS A NULL POINTER, THIS METHOD KNOWS WHAT TO DO WITH IT
METHOD2
CATCH (NULL POINTER EXCEPTION)
EXCEPTION GOES ONE LEVEL UP THE STACK
METHOD1
SOMETHING GOES WRONG!
EXCEPTION GETS THROWN - EITHER NULLPOINTEREXCEPTION OR FILEIOEXCEPTION