TRY/CATCH WITH EXCEPTIONS

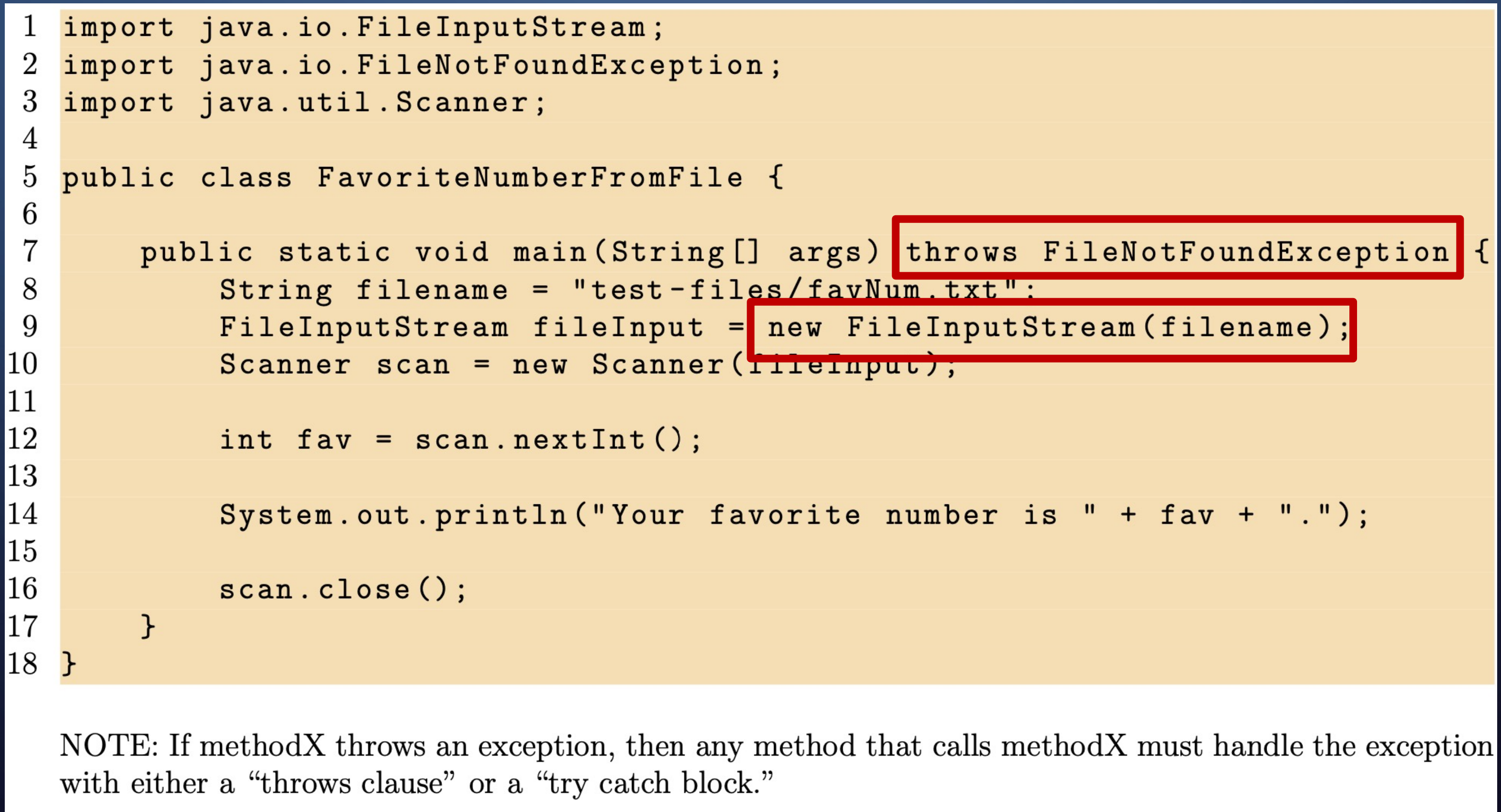
zyBook 10.1, zyBook 10.2
FileNotFoundException API, Oracle - Java Tutorial: Exceptions

RECAP – EXCEPTIONS

- Exceptions are errors that prevent a program from continuing normal execution
  - Exceptions are "thrown" by the program
  - We can "catch" exceptions to handle them
- What is a FileNotFoundException?
  - A FileNotFoundException is thrown because the program is worried that the file might not exist on the file system.
  - Unlike the exceptions we have discussed previously, FileNotFoundException is a checked exception.
    - Checked Exception → an exception that must be caught or specifically declared in the header of the method that might generate it.
  - Compiler "checks" that we are aware of exception.

WHAT IS A THROWS CLAUSE?

A throws clause is a declaration that a method will not attempt to handle a particular type of exception. It is explicit acknowledgement that a statement in the method may throw the exception.



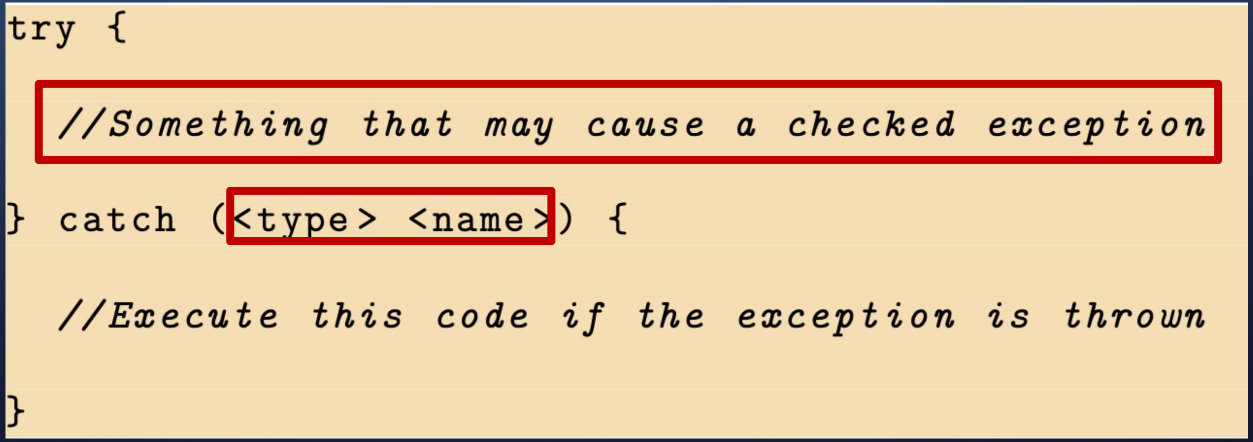
```java
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class FavoriteNumberFromFile {

    public static void main(String[] args) throws FileNotFoundException {
        String filename = "test-files/favNum.txt";
        FileInputStream fileInput = new FileInputStream(filename);
        Scanner scan = new Scanner(fileInput);

        int fav = scan.nextInt();

        System.out.println("Your favorite number is " + fav + ".");

        scan.close();
    }
}
```

NOTE: If methodX throws an exception, then any method that calls methodX must handle the exception with either a “throws clause” or a “try catch block.”

CATCHING EXCEPTIONS

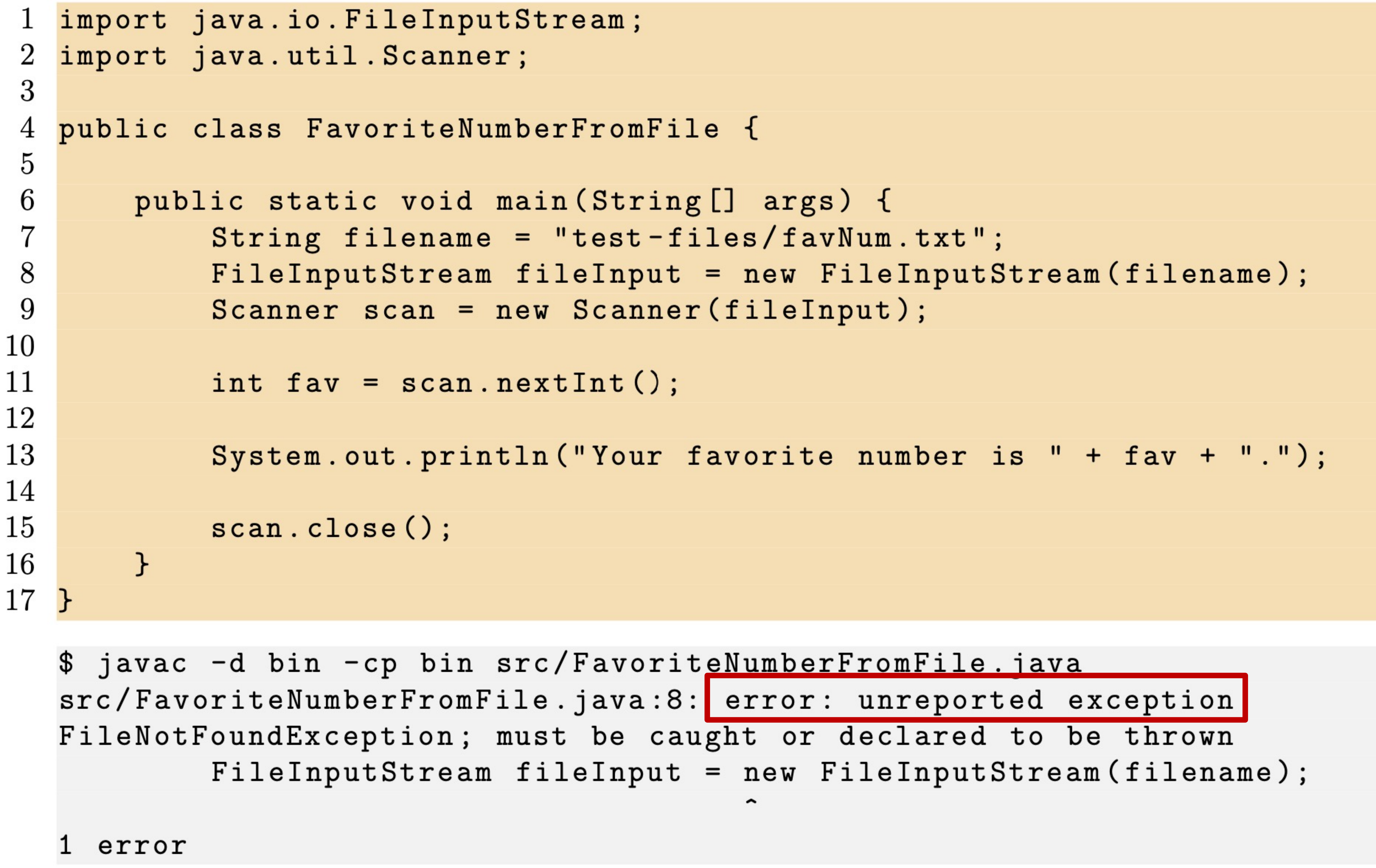
Using a throws clause leads to compiling code, but not a robust program.

- We want the ability to continue running your program even if the file does not exist!
- We want the ability to end the program without throwing an exception.



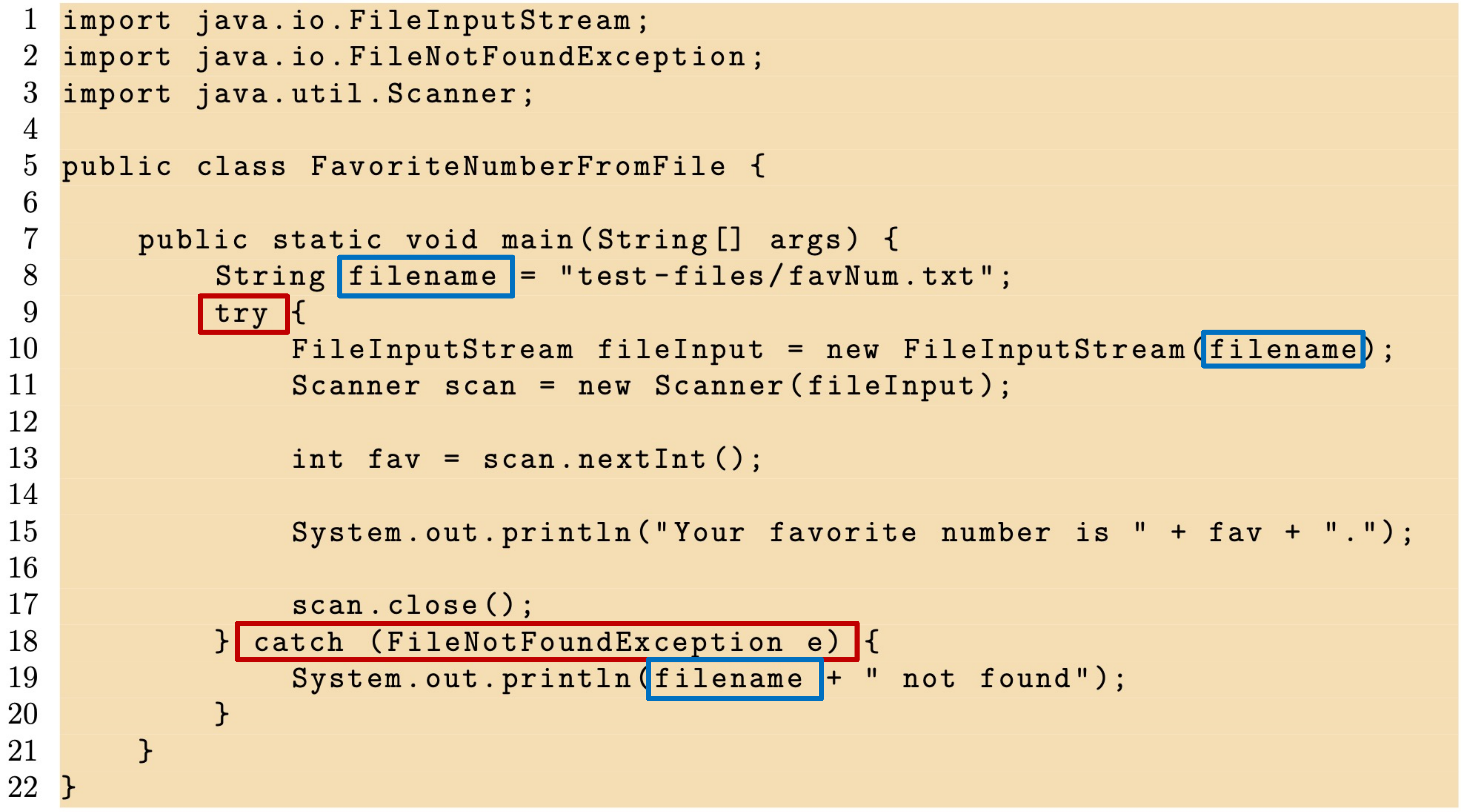
```
try {
  //Something that may cause a checked exception
} catch (<type> <name>) {
  //Execute this code if the exception is thrown
}
```

UPDATING FAVORITENUMBERFROMFILE TO CATCH EXCEPTION



```java
import java.io.FileInputStream;
import java.util.Scanner;

public class FavoriteNumberFromFile {

    public static void main(String[] args) {
        String filename = "test-files/favNum.txt";
        FileInputStream fileInput = new FileInputStream(filename);
        Scanner scan = new Scanner(fileInput);

        int fav = scan.nextInt();

        System.out.println("Your favorite number is " + fav + ".");

        scan.close();
    }
}
```

```
$ javac -d bin -cp bin src/FavoriteNumberFromFile.java
src/FavoriteNumberFromFile.java:8: error: unreported exception
FileNotFoundException; must be caught or declared to be thrown
        FileInputStream fileInput = new FileInputStream(filename);
                                    ^
1 error
```



```java
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class FavoriteNumberFromFile {

    public static void main(String[] args) {
        String filename = "test-files/favNum.txt";
        try {
            FileInputStream fileInput = new FileInputStream(filename);
            Scanner scan = new Scanner(fileInput);

            int fav = scan.nextInt();

            System.out.println("Your favorite number is " + fav + ".");

            scan.close();
        } catch (FileNotFoundException e) {
            System.out.println(filename + " not found");
        }
    }
}
```