```
import java.io.FileNotFoundException;

public class SuspiciousClass {

    public void doSomething() throws
                    FileNotFoundException {

        throw new FileNotFoundException();
    }

}
```

throw means I am forcing an exception to happen

throws means this method MAY throw an exception

```
import java.io.FileNotFoundException;

public class SuspiciousClass {

  public void doSomething() throws
                FileNotFoundException {

     throw new FileNotFoundException();
  }
```

```
public class MyMainClass {

   public static void main(String[] args)  {
      SuspiciousClass test = new
                    SuspciousClass();
      test.doSomething();
   }
}
```

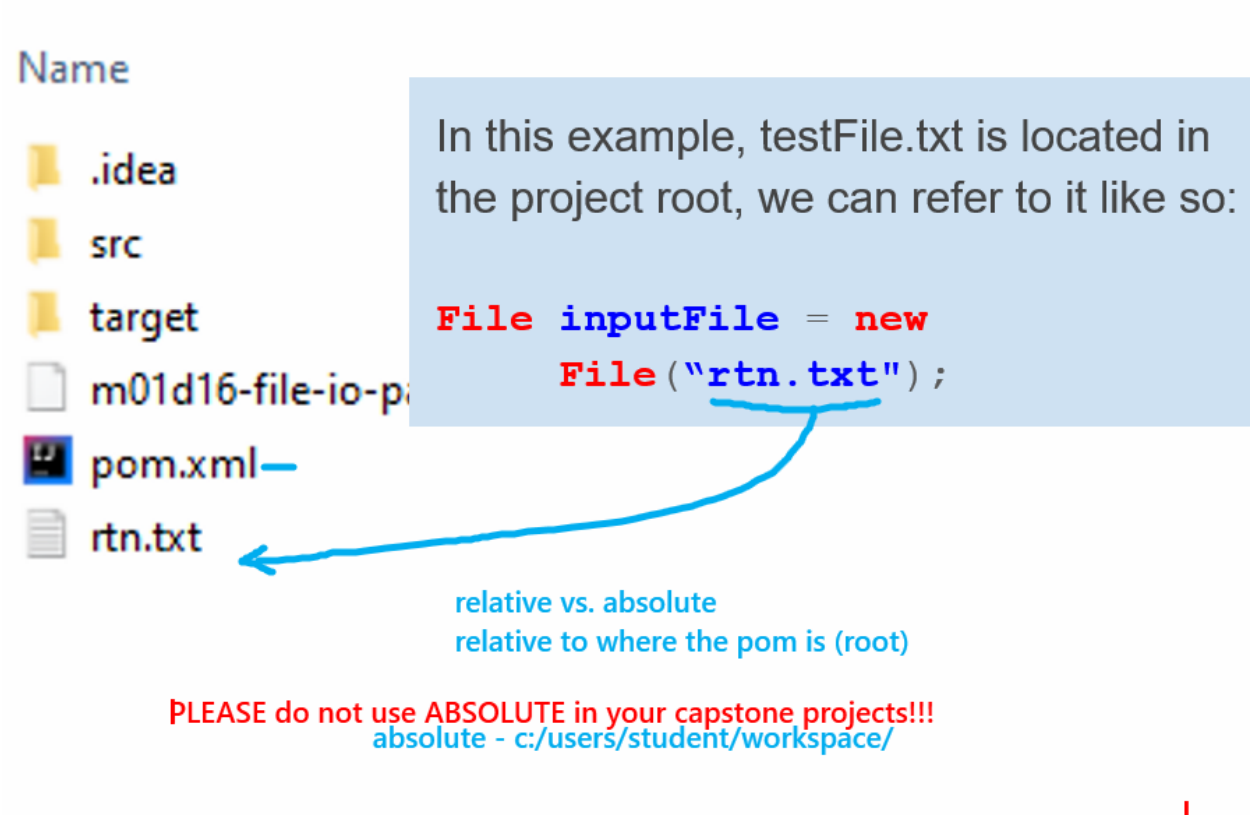
throw means I am forcing an exception to happen

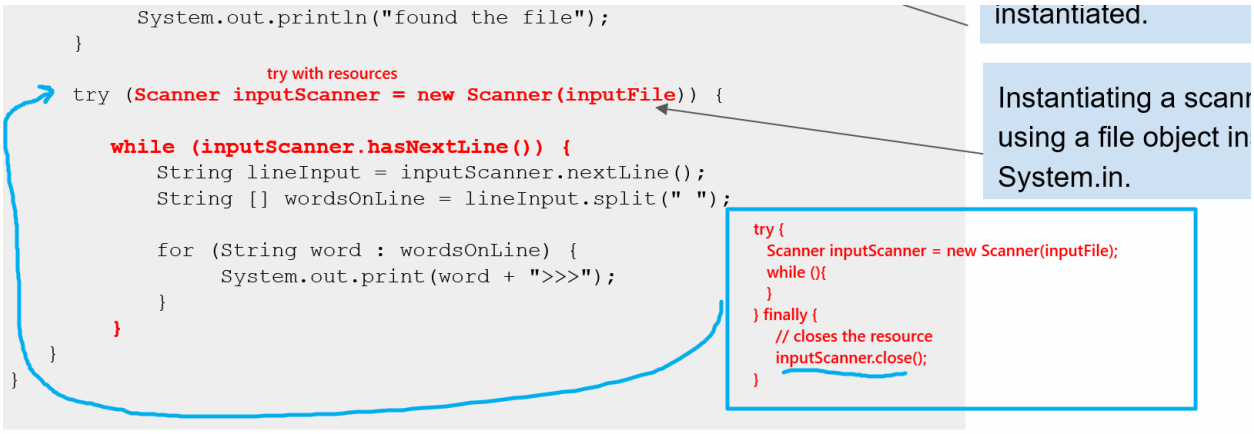
```
if (blah) {
   throw new FileNotFoundException();
```

throws means this method MAY throw an exception

throws the method has the ability to throw an exception
that we are going to pass up the program stack --
it is not handled right here in this method

Two choices if a method will throw an exception(either intentionally or a checked exception)
1. you either add throws on the method header
2. you add try catch language


Name
.idea
src
target
m01d16-file-io-p
pom.xml
rtn.txt
In this example, testFile.txt is located in the project root, we can refer to it like so:
File inputFile = new File("rtn.txt");
relative vs. absolute
relative to where the pom is (root)
PLEASE do not use ABSOLUTE in your capstone projects!!!
absolute - c:/users/student/workspace/


System.out.println("found the file");
}
try with resources
try (Scanner inputScanner = new Scanner(inputFile)) {
while (inputScanner.hasNextLine()) {
String lineInput = inputScanner.nextLine();
String [] wordsOnLine = lineInput.split(" ");
for (String word : wordsOnLine) {
System.out.print(word + ">>>");
}
}
}
}
instantiated.
Instantiating a scanr
using a file object in
System.in.
try {
Scanner inputScanner = new Scanner(inputFile);
while (){
}
} finally {
// closes the resource
inputScanner.close();
}