

● THROWS

➤ Throws Example , And when you hover over the error you get options like below.

The screenshot shows the Eclipse IDE with a Java file named `Main.java`. The code is as follows:

```
1 package org.studyeasy;
2
3 import java.io.FileReader;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Main main = new Main();
9         main.doSomething();
10    }
11
12    public void doSomething() {
13        FileReader in = new FileReader("text.txt");
14        System.out.println("Do something");
15    }
16 }
```

The IDE shows a red squiggly line under the `FileReader` constructor on line 12, indicating an unresolved compilation problem. The console output at the bottom shows the following error:

```
<terminated> Main (3) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Unhandled exception type FileNotFoundExcep...
    at org.studyeasy.Main.doSomething(Main.java:12)
    at org.studyeasy.Main.main(Main.java:9)
```

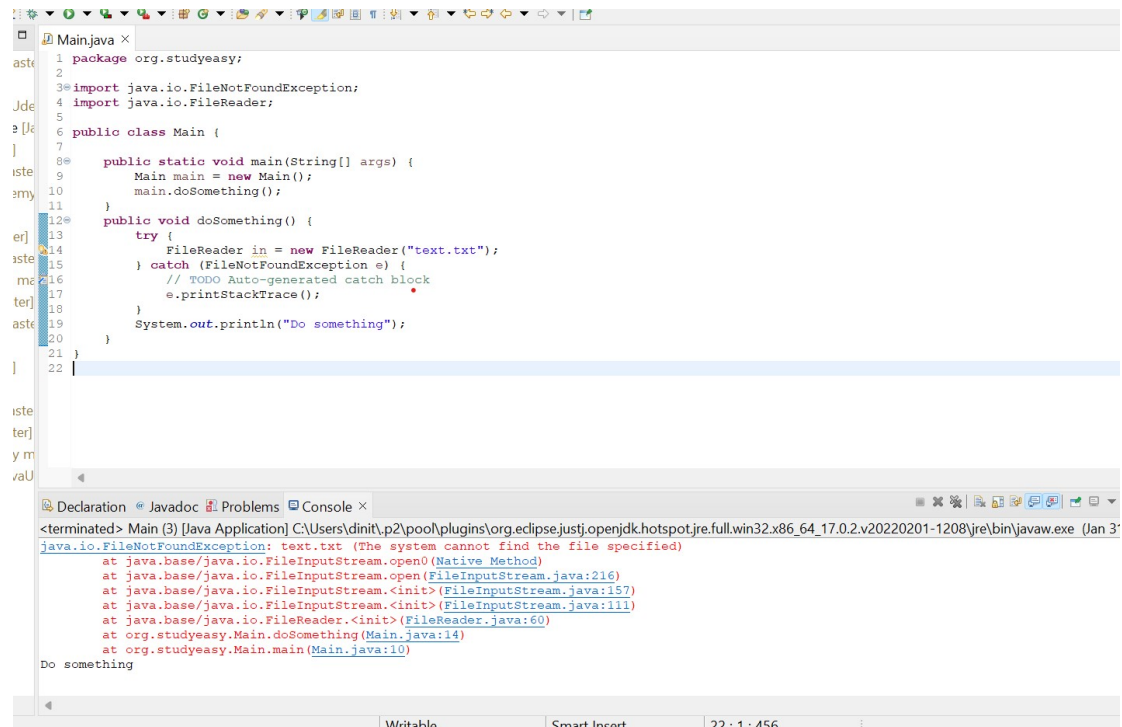
The screenshot shows the Eclipse IDE with the same `Main.java` file. A hover tooltip is displayed over the `FileReader` constructor on line 12, showing the following information:

- Unhandled exception type `FileNotFoundExcep...`
- 2 quick fixes available:
 - [Add throws declaration](#)
 - [Surround with try/catch](#)

The console output at the bottom shows the same error as the first screenshot:

```
<terminated> Main (3) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Jan 31, 21
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Unhandled exception type FileNotFoundExcep...
    at org.studyeasy.Main.doSomething(Main.java:12)
    at org.studyeasy.Main.main(Main.java:9)
```

Case 1: Surround with try/catch



```
1 package org.studyeasy;
2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         Main main = new Main();
10        main.doSomething();
11    }
12    public void doSomething() {
13        try {
14            FileReader in = new FileReader("text.txt");
15        } catch (FileNotFoundException e) {
16            // TODO Auto-generated catch block
17            e.printStackTrace();
18        }
19        System.out.println("Do something");
20    }
21 }
22
```

Declaration Javadoc Problems Console

<terminated> Main (3) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (Jan 3'

java.io.FileNotFoundException: text.txt (The system cannot find the file specified)

at java.base/java.io.FileInputStream.open0(Native Method)

at java.base/java.io.FileInputStream.open(FileInputStream.java:216)

at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)

at java.base/java.io.FileReader.<init>(FileReader.java:111)

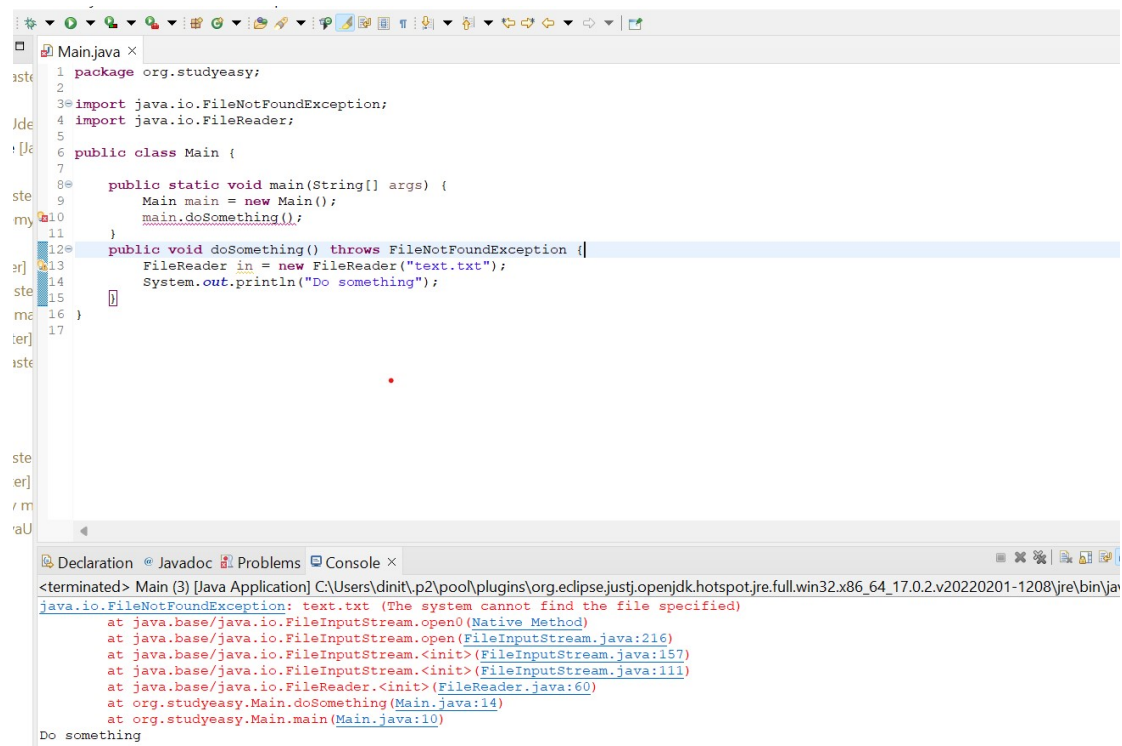
at java.base/java.io.FileReader.<init>(FileReader.java:60)

at org.studyeasy.Main.doSomething(Main.java:14)

at org.studyeasy.Main.main(Main.java:10)

Do something

Case 2 : Add throws declaration



```
1 package org.studyeasy;
2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         Main main = new Main();
10        main.doSomething();
11    }
12    public void doSomething() throws FileNotFoundException {
13        FileReader in = new FileReader("text.txt");
14        System.out.println("Do something");
15    }
16 }
17
```

Declaration Javadoc Problems Console

<terminated> Main (3) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\ja

java.io.FileNotFoundException: text.txt (The system cannot find the file specified)

at java.base/java.io.FileInputStream.open0(Native Method)

at java.base/java.io.FileInputStream.open(FileInputStream.java:216)

at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)

at java.base/java.io.FileReader.<init>(FileReader.java:111)

at java.base/java.io.FileReader.<init>(FileReader.java:60)

at org.studyeasy.Main.doSomething(Main.java:14)

at org.studyeasy.Main.main(Main.java:10)

Do something

In Java, the throws keyword is used in the method signature to declare that a method may throw certain exceptions. This keyword is part of the method declaration and is used to indicate that the method might generate exceptions of a particular type during its execution.

```
public void exampleMethod() throws SomeException {
    // Method implementation that may throw SomeException
}
```

- Throws keyword is used in order to throw an exception.
- If we try catch statement - in another block like below, then we get output like below.

```
1 package org.studyeasy;
2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         Main main = new Main();
10        try {
11            main.doSomething();
12        } catch (FileNotFoundException e) {
13            // TODO Auto-generated catch block
14            e.printStackTrace();
15        }
16    }
17
18    public void doSomething() throws FileNotFoundException {
19        FileReader in = new FileReader("text.txt");
20        System.out.println("Do something");
21    }
22 }
```

Declaration @ Javadoc Problems Console x

```
<terminated> Main (3) [Java Application] C:\Users\dinit\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Jan 31
java.io.FileNotFoundException: text.txt (The system cannot find the file specified)
    at java.base/java.io.FileInputStream.open0(Native Method)
    at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
    at java.base/java.io.FileReader.<init>(FileReader.java:111)
    at org.studyeasy.Main.doSomething(Main.java:18)
    at org.studyeasy.Main.main(Main.java:11)
```

You won't get the output "Do something"

- If you don't know how to handle the exception by yourself and the exception handling will be unique for every case. It is better idea to throw the exception, so whenever the user will use the method, the user have to handle the exception according to the situation.

The screenshot shows the Eclipse IDE with a Java project. The editor displays `Main.java` with the following code:

```
1 package org.studyeasy;
2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         Main main = new Main();
10        try {
11            main.doSomething();
12        } catch (FileNotFoundException e) {
13            // TODO Auto-generated catch block
14            e.printStackTrace();
15        }
16    }
17    public void doSomething() throws FileNotFoundException {
18        FileReader in = new FileReader("text.txt");
19        System.out.println("Do something");
20    }
21 }
22
```

The `doSomething` method is highlighted in blue. The console at the bottom shows the following stack trace:

```
<terminated> Main (3) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe (Ja
java.io.FileNotFoundException: text.txt (The system cannot find the file specified)
    at java.io.FileInputStream.open0(Native Method)
    at java.io.FileInputStream.open(FileInputStream.java:216)
    at java.io.FileInputStream.<init>(FileInputStream.java:157)
    at java.io.FileInputStream.<init>(FileInputStream.java:111)
    at java.io.FileReader.<init>(FileReader.java:60)
    at org.studyeasy.Main.doSomething(Main.java:18)
    at org.studyeasy.Main.main(Main.java:11)
```

An arrow points from the `throws FileNotFoundException` line in the code to the `FileNotFoundException` in the stack trace.

User is forced to handle the exception as the throws keyword forces it.