

Run time exception-

workspace1 - May2024JavaSessions/src/ExceptionHandling/TestException

```

Employee.java *TestException.java
1 package ExceptionHandling;
2
3 public class TestException {
4
5
6   public static void main(String[] args) {
7
8     int i = 9/0; //RunTime Exception
9
10
  
```

Compile time exception-

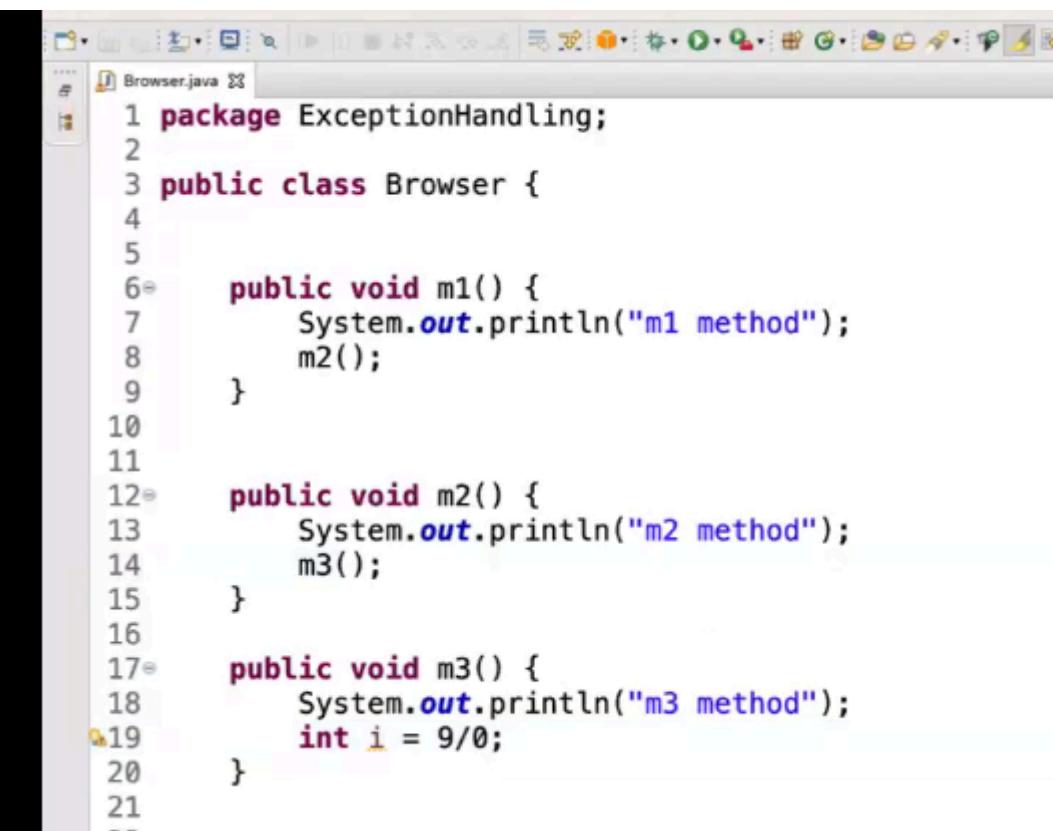
```

15 try {
16   FileInputStream ip = new FileInputStream("test.xls");//CT
17 } catch (FileNotFoundException e) {
18   e.printStackTrace()
19 }
20
21
22 try {
23   Thread.sleep(5000);
24 } catch (InterruptedException e) {
25   e.printStackTrace();
26 } //5 secs --- CT
27
  
```

Errors are run time exceptions.

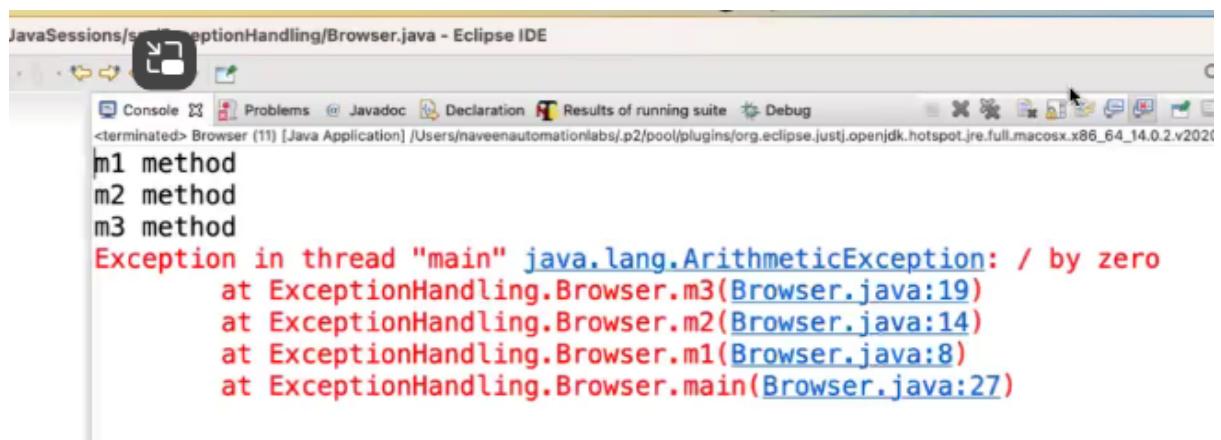
Compile time exception known as checked exception.

Run time exception known as unchecked exception.



The screenshot shows a Java IDE interface with the file 'Browser.java' open. The code defines a class 'Browser' with three methods: m1(), m2(), and m3(). The m3() method contains a division by zero error. The IDE's status bar at the bottom indicates 'Java code completion'.

```
1 package ExceptionHandling;
2
3 public class Browser {
4
5
6     public void m1() {
7         System.out.println("m1 method");
8         m2();
9     }
10
11
12     public void m2() {
13         System.out.println("m2 method");
14         m3();
15     }
16
17     public void m3() {
18         System.out.println("m3 method");
19         int i = 9/0;
20     }
21
22
23
24     public static void main(String[] args) {
25         Browser br = new Browser();
26         br.m1();|
27
28
29     }
30
31 }
```

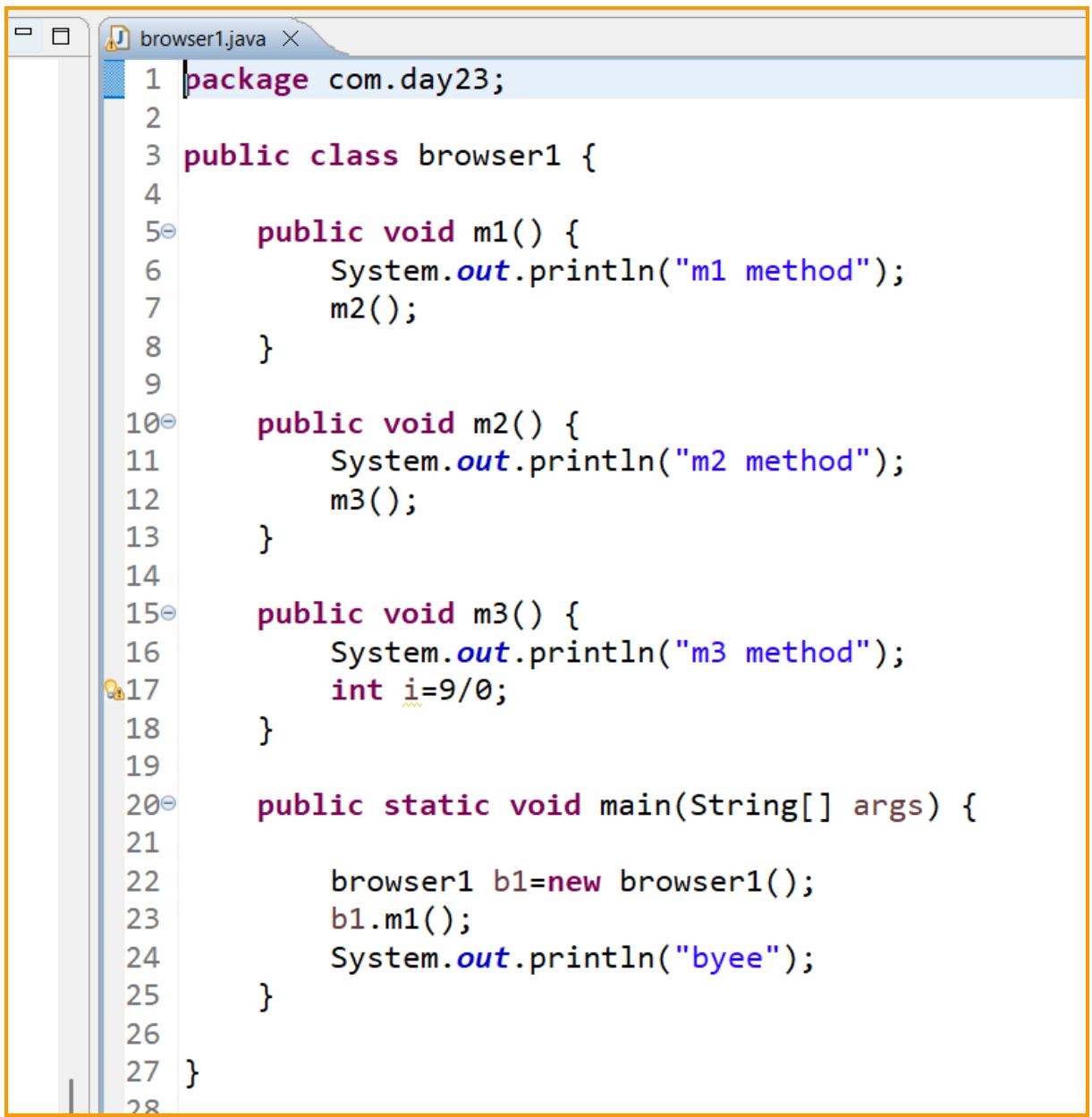


```

JavaSessions/seleniumExceptionHandling/ExceptionHandling/Browser.java - Eclipse IDE
Console Problems Javadoc Declaration Results of running suite Debug
<terminated> Browser (11) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200520-1245
m1 method
m2 method
m3 method
Exception in thread "main" java.lang.ArithmetricException: / by zero
    at ExceptionHandling.Browser.m3(Browser.java:19)
    at ExceptionHandling.Browser.m2(Browser.java:14)
    at ExceptionHandling.Browser.m1(Browser.java:8)
    at ExceptionHandling.Browser.main(Browser.java:27)

```

paste browser1-



```

browser1.java
1 package com.day23;
2
3 public class browser1 {
4
5     public void m1() {
6         System.out.println("m1 method");
7         m2();
8     }
9
10    public void m2() {
11        System.out.println("m2 method");
12        m3();
13    }
14
15    public void m3() {
16        System.out.println("m3 method");
17        int i=9/0;
18    }
19
20    public static void main(String[] args) {
21
22        browser1 b1=new browser1();
23        b1.m1();
24        System.out.println("bye");
25    }
26
27 }
28

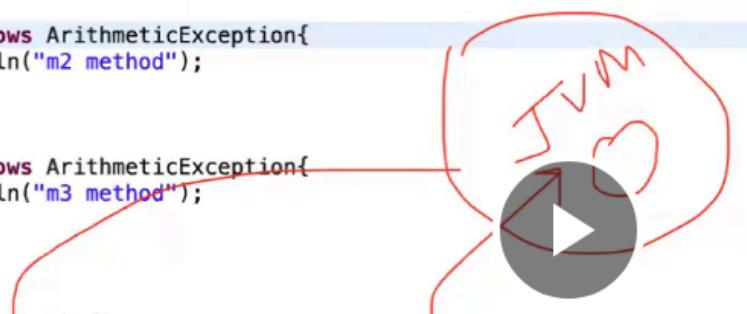
```

```

26
27 }
28
29 //m1 method
30 //m2 method
31 //m3 method
32 //Exception in thread "main" java.lang.ArithmetricException: / by zero
33 // at com.day23.browser1.m3(browser1.java:17)
34 // at com.day23.browser1.m2(browser1.java:12)
35 // at com.day23.browser1.m1(browser1.java:7)
36 // at com.day23.browser1.main(browser1.java:23)
37
38
39

```

Throws-



```

1 package ExceptionHandling;
2
3 public class Browser {
4
5
6     public void m1() throws ArithmetricException{
7         System.out.println("m1 method");
8         m2();
9     }
10
11
12     public void m2() throws ArithmetricException{
13         System.out.println("m2 method");
14         m3();
15     }
16
17     public void m3() throws ArithmetricException{
18         System.out.println("m3 method");
19         int i = 9/0;
20     }
21
22
23     //caller method: user method
24     public static void main(String[] args) throws ArithmetricException{
25
26         Browser br = new Browser();
27         br.m1();
28
29         System.out.println("bye!!!");
30
31     }
32
33 }
34
35

```



```

sessions/src/ExceptionHandling/ExceptionHandling.java - Eclipse IDE
Console Problems Javadoc Declaration Results of running suite Debug
<terminated> Browser (11) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200129-1245
m1 method
m2 method
m3 method
Exception in thread "main" java.lang.ArithmetricException: / by zero
    at ExceptionHandling.Browser.m3(Browser.java:19)
    at ExceptionHandling.Browser.m2(Browser.java:14)
    at ExceptionHandling.Browser.m1(Browser.java:8)
    at ExceptionHandling.Browser.main(Browser.java:27)

```

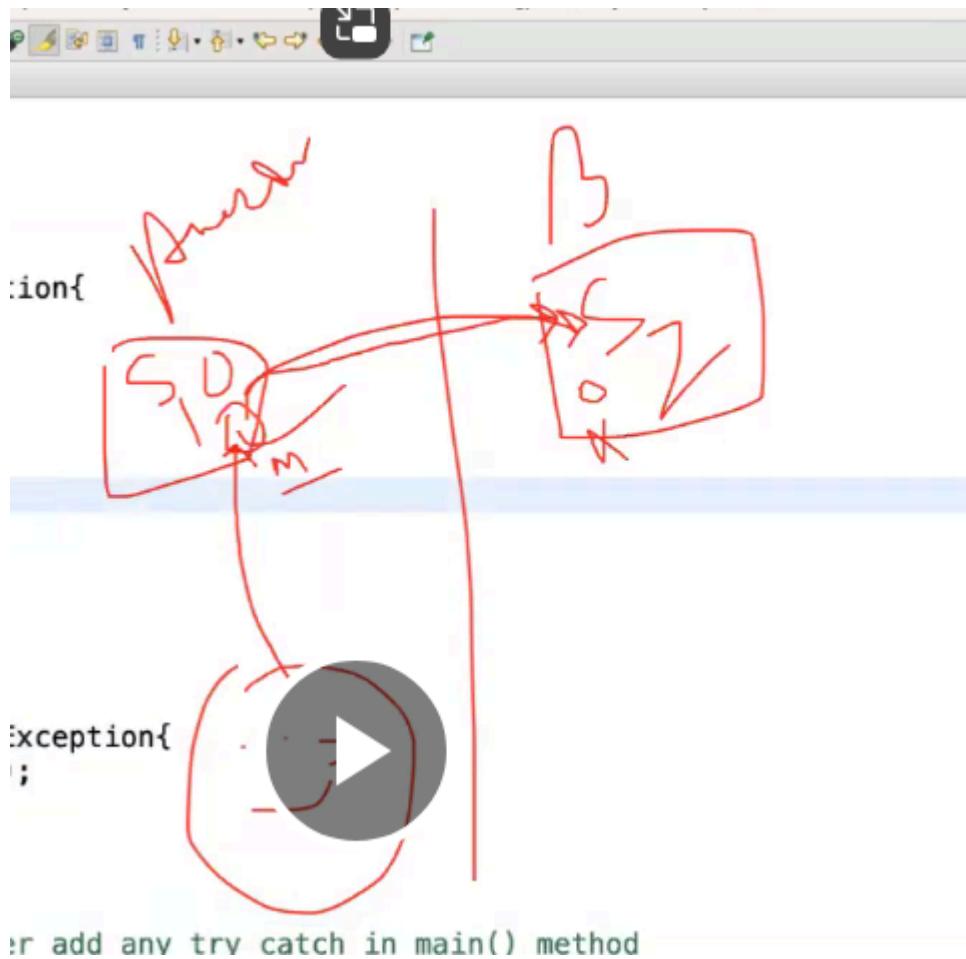
Throws – not handles exception. 21.50

```

22 //caller method: user method: we should never add any try catch in main() method
23 public static void main(String[] args) throws ArithmetricException{
24
25     Browser br = new Browser();
26     br.m1();
27
28     System.out.println("bye!!");
29
30
31
32 }

```

Our application should handle exception-
 Example amazon website some functionality.



if add any try catch in main() method

This is true when we talk with external parties
(amazon website shopping) also.

Throws and also handle exception- 21.32



```

1 package ExceptionHandling;
2
3 public class Browser {
4
5     // amazon app
6     public void search() {
7         System.out.println("m1 method");
8         cart();
9     }
10
11    // amazon app
12    public void cart() {
13        System.out.println("m2 method");
14        try {
15            bankPayment();
16        } catch (ArithmetricException e) {
17            e.printStackTrace();
18            System.out.println("some AE is coming...");
19        }
20    }
21
22    // HDFC bank application: 3rd party system
23    public void bankPayment() throws ArithmetricException {
24        System.out.println("bankPayment method");
25        int i = 9 / 0;
26    }
27
28    // caller method: user method: we should never add any try catch in main()
29    // method
30    public static void main(String[] args) {
31
32        Browser br = new Browser();
33        br.search();
34
35        System.out.println("bye!!!");
36
37    }

```

Console Problems Javadoc Declaration Results of running suite Debug

<terminated> Browser (11) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815

```

m1 method
m2 method
bankPayment method
java.lang.ArithmetricException: / by zero
    at ExceptionHandling.Browser.bankPayment(Browser.java:25)
    at ExceptionHandling.Browser.cart(Browser.java:15)
    at ExceptionHandling.Browser.search(Browser.java:8)
    at ExceptionHandling.Browser.main(Browser.java:33)
some AE is coming...
bye!!!

```

paste browser3-

```

browser3.java X
1 package com.day23;
2
3 public class browser3 {
4
5     //throws and handling exception together.
6
7     //amazon app search
8     public void search() {
9         System.out.println("search method of amazon");
10        String s1=cart();
11        System.out.println(s1);
12    }
13
14     public String cart() {
15         System.out.println("Cart method of amazon");
16
17         try {
18             bankpayment();
19         }catch(ArithmeticException e) {
20             e.printStackTrace();
21             System.out.println("arithmetic exception is coming");
22         }
23         return "tiger";
24     }
25
26     public void bankpayment() throws ArithmeticException {
27         System.out.println("bank payment method");
28         int i=9/0;
29     }
30
31     public static void main(String[] args) throws ArithmeticException {
32
33         browser3 b1=new browser3();
34         b1.search();
35         System.out.println("byee");
36     }
37
38 }
39
40 //search method of amazon
41 //Cart method of amazon
42 //bank payment method
43 //java.lang.ArithmaticException: / by zero
44 //  at com.day23.browser3.bankpayment(browser3.java:28)
45 //  at com.day23.browser3.cart(browser3.java:18)
46 //  at com.day23.browser3.search(browser3.java:10)
47 //  at com.day23.browser3.main(browser3.java:34)
48 //arithmetic exception is coming
49 //tiger
50 //byee
51

```

Remove throws-



```

1 package ExceptionHandling;
2
3 public class Browser {
4
5     // amazon app
6     public void search() {
7         System.out.println("m1 method");
8         cart();
9     }
10
11    // amazon app
12    public void cart() {
13        System.out.println("m2 method");
14        try {
15            bankPayment();
16        } catch (ArithmetricException e) {
17            e.printStackTrace();
18            System.out.println("some AE is coming...");
19        }
20    }
21
22    // HDFC bank application: 3rd party system
23    public void bankPayment(){
24        System.out.println("bankPayment method");
25        int i = 9 / 0;
26    }
27
28    // caller method: user method: we should never add any try catch in main()
29    // method
30    public static void main(String[] args) {
31
32        Browser br = new Browser();
33        br.search();
34
35        System.out.println("bye!!");
36    }
37
38
39

```

JavaSession/ExceptionHandling/Browser.java - Eclipse IDE

Console Problems Javadoc Declaration Results of running suite Debug

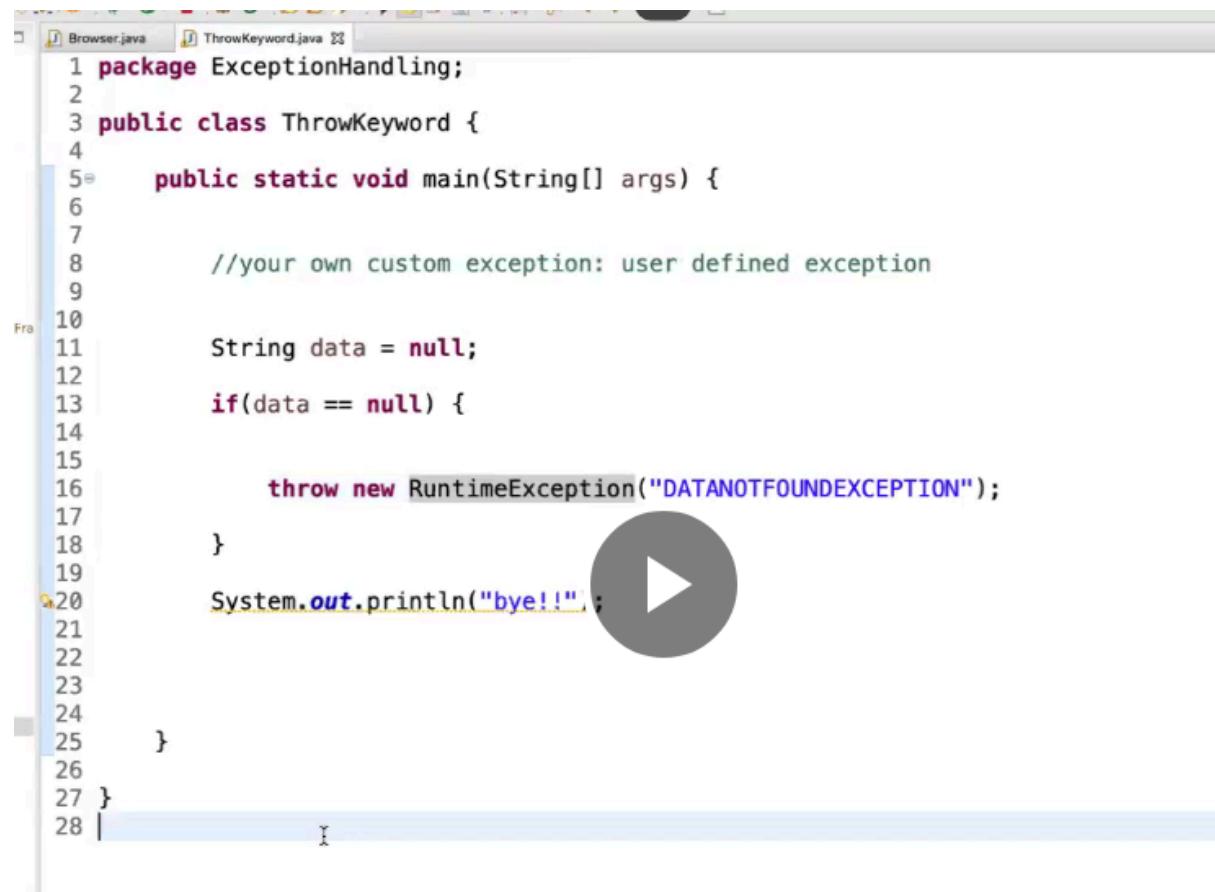
<terminated> Browser (11) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20140710-0945

```

m1 method
m2 method
bankPayment method
java.lang.ArithmetricException: / by zero
    at ExceptionHandling.Browser.bankPayment(Browser.java:25)
    at ExceptionHandling.Browser.cart(Browser.java:15)
    at ExceptionHandling.Browser.search(Browser.java:8)
    at ExceptionHandling.Browser.main(Browser.java:33)
some AE is coming...
bye!!

```

Throw keyword-



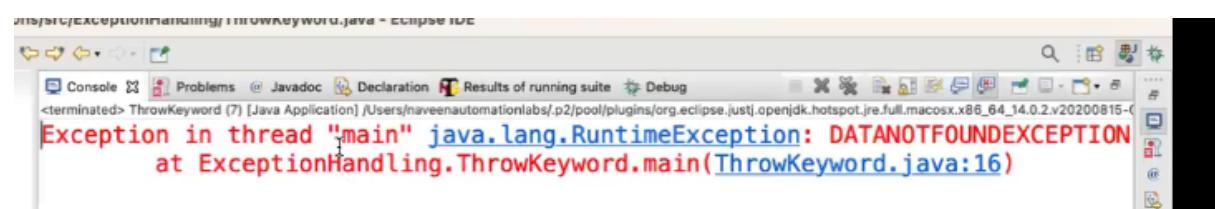
The screenshot shows the Eclipse IDE interface with the code editor open. The code is as follows:

```

1 package ExceptionHandling;
2
3 public class ThrowKeyword {
4
5     public static void main(String[] args) {
6
7         //your own custom exception: user defined exception
8
9
10        String data = null;
11
12        if(data == null) {
13
14
15            throw new RuntimeException("DATANOTFOUNDEXCEPTION");
16
17        }
18
19        System.out.println("bye!!");
20
21
22
23
24    }
25
26
27 }
28

```

A large play button icon is overlaid on the code editor area.



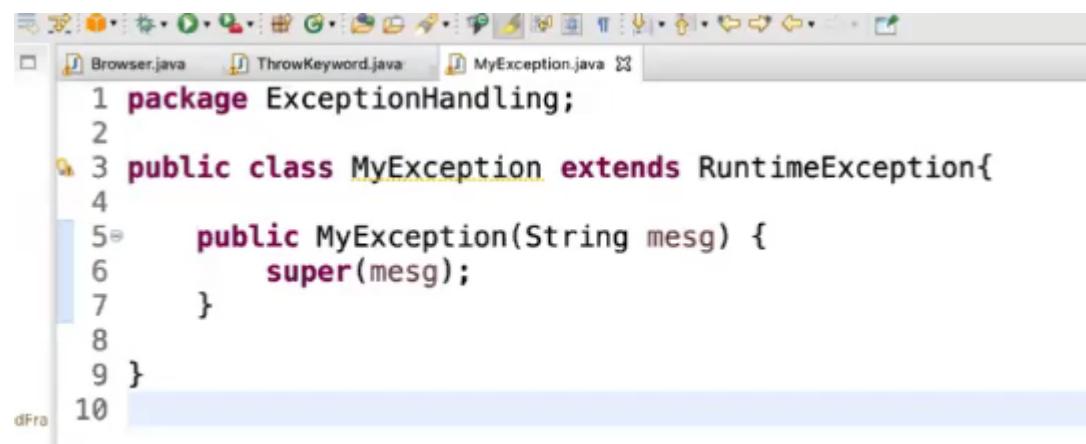
The screenshot shows the Eclipse IDE's Console view with the following output:

```

Exception in thread "main" java.lang.RuntimeException: DATANOTFOUNDEXCEPTION
at ExceptionHandling.ThrowKeyword.main(ThrowKeyword.java:16)

```

Own exception creation-



The screenshot shows the Eclipse IDE interface with the code editor open. The code is as follows:

```

1 package ExceptionHandling;
2
3 public class MyException extends RuntimeException{
4
5     public MyException(String mesg) {
6         super(mesg);
7     }
8
9 }
10

```

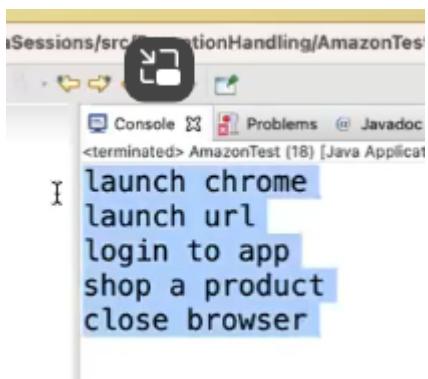
Amazon-



```

1 package ExceptionHandling;
2
3 public class AmazonTest {
4
5     public static void main(String[] args) {
6
7         String browser = "chrome";
8
9         switch (browser) {
10            case "chrome":
11                System.out.println("launch chrome");
12                break;
13            case "firefox":
14                System.out.println("launch ff");
15                break;
16            case "safari":
17                System.out.println("launch safari");
18                break;
19            case "edge":
20                System.out.println("launch edge");
21                break;
22
23            default:
24                System.out.println("plz pass the right browser....");
25                break;
26        }
27
28
29
30        System.out.println("launch url");
31        System.out.println("login to app");
32        System.out.println("shop a product");
33        System.out.println("close browser");
34    }
35
36 }
37

```



```

ISessions/src/ExceptionHandling/AmazonTest
[Run] [Stop] [Console]
Console Problems Javadoc
<terminated> AmazonTest (18) [Java Application]
I launch chrome
I launch url
I login to app
I shop a product
I close browser

```

Throw own exception-

```

24     default:
25         System.out.println("plz pass the right browser...." + browser);
26         throw new MyException("====WRONG BROWSER PASSED====");
27     }
28

```



paste amazontest2-

The screenshot shows the Eclipse IDE interface with the code editor open to the file 'amazontest2.java'. The code implements a switch statement to determine which browser to print based on the input argument.

```
1 package com.day23;
2
3 public class amazontest2 {
4
5     public static void main(String[] args) {
6
7         String browser="chrome";
8
9         switch(browser) {
10
11             case "chrome":
12                 System.out.println("Chrome browser");
13                 break;
14
15             case "safari":
16                 System.out.println("safari browser");
17                 break;
18
19             case "firefox":
20                 System.out.println("firefox browser");
21                 break;
22
23             case "edge":
24                 System.out.println("edge browser");
25                 break;
26         }
27     }
28 }
```

```
24         System.out.println("edge browser");
25         break;
26
27     default:
28         System.out.println("please pass right browser");
29         throw new myexception("wrong browser is passed");
30         //throw and break cannot come together.
31         break; //Unreachable code
32     }
33
34
35     System.out.println("byee");
36 }
37
38 }
39
40 //Chrome browser
41 //byee
42
43
```

paste amazontest4-

```

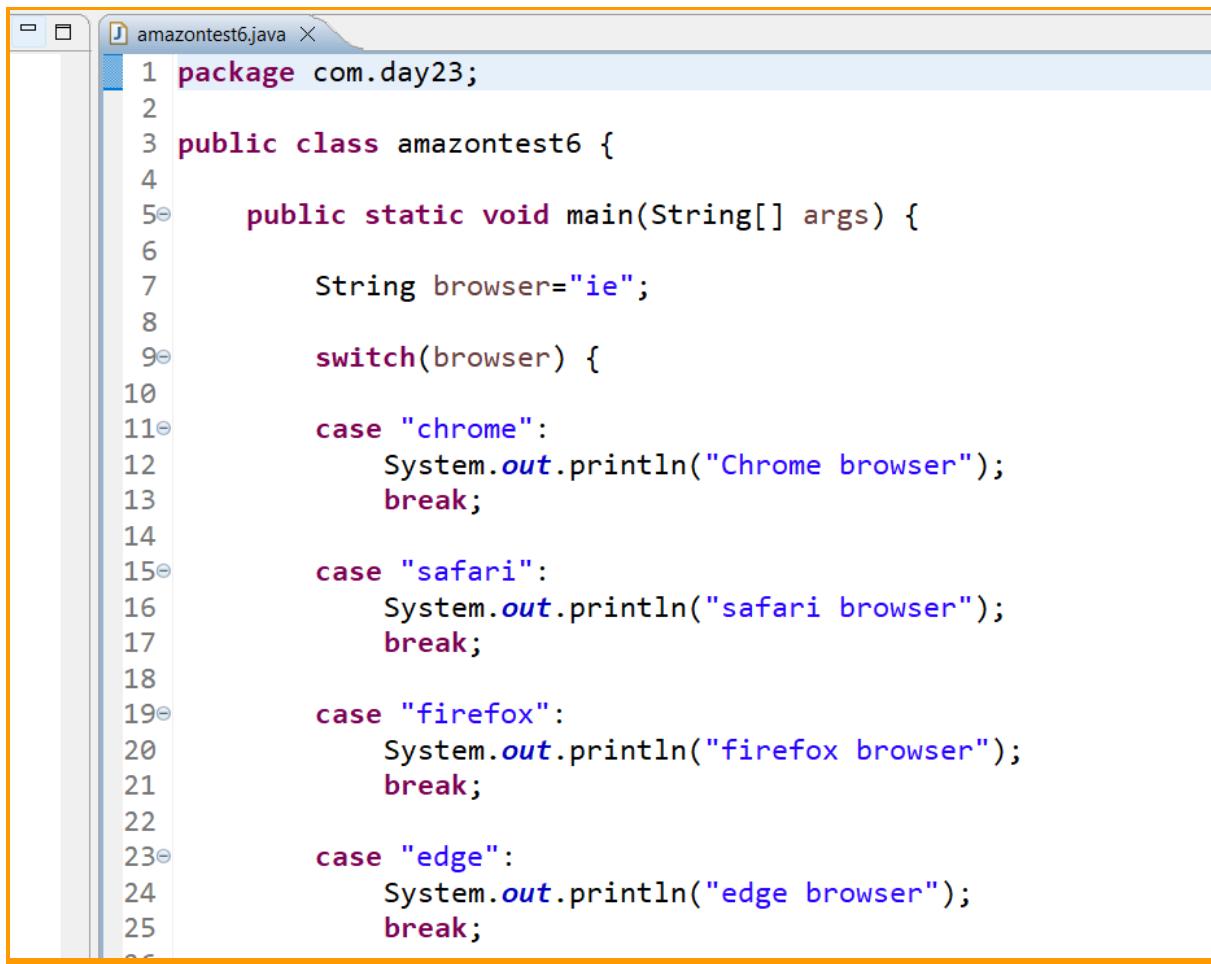
amazontest4.java X
1 package com.day23;
2
3 public class amazontest4 {
4
5     public static void main(String[] args) {
6
7         String browser="chrome";
8
9         switch(browser) {
10
11             case "chrome":
12                 System.out.println("Chrome browser");
13                 break;
14
15             case "safari":
16                 System.out.println("safari browser");
17                 break;
18
19             case "firefox":
20                 System.out.println("firefox browser");
21                 break;
22
23             case "edge":
24                 System.out.println("edge browser");
25                 break;
26
27                 System.out.println("edge browser");
28                 break;
29
30             default:
31                 System.out.println("please pass right browser");
32                 throw new myexception("wrong browser is passed");
33                 //throw and break cannot come together.
34                 break; //Unreachable code
35
36         System.out.println("byee");
37     }
38 }
39
40 //Chrome browser
41 //byee
42
43

```

Throw inside try catch- 59.00

throw in this way will also not handle exception.
see catch block will not run only.

paste amazontest6-



```
amazontest6.java
1 package com.day23;
2
3 public class amazontest6 {
4
5     public static void main(String[] args) {
6
7         String browser="ie";
8
9         switch(browser) {
10
11             case "chrome":
12                 System.out.println("Chrome browser");
13                 break;
14
15             case "safari":
16                 System.out.println("safari browser");
17                 break;
18
19             case "firefox":
20                 System.out.println("firefox browser");
21                 break;
22
23             case "edge":
24                 System.out.println("edge browser");
25                 break;
26
27         }
28     }
29 }
```

```

24     System.out.println("edge browser");
25     break;
26
27     default:
28         System.out.println("please pass right browser");
29         //throw can be inside try also.
30         //but catch block wont run as exception not catched and handled.
31         try {
32             throw new myexception("wrong browser is passed");
33         }
34         catch(Exception e) {
35             System.out.println(e);
36         }
37     }
38
39
40     System.out.println("byee");
41 }
42
43 }
44
45 //please pass right browser
46 //com.day23.myexception: wrong browser is passed
47 //byee
48

```

```

22
23     default:
24         System.out.println("plz pass the right browser...." + browser);
25         try {
26             throw new MyException("====WRONG BROWSER PASSED====");
27         } catch (Exception e) {
28
29         }
30     }
31

```

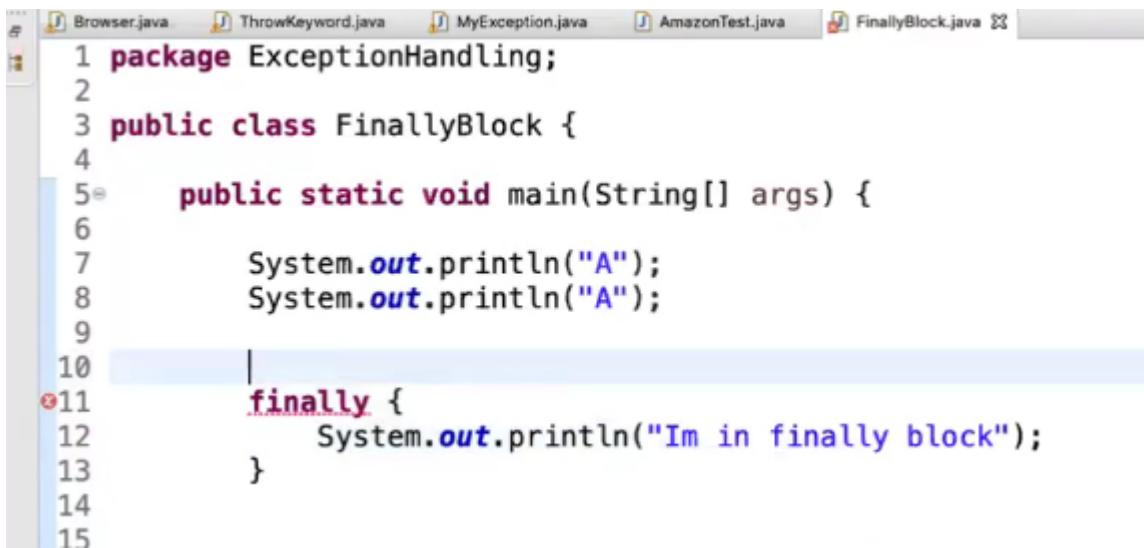


Finally block-

Always runs irrespective of exception or not.

Finally cannot come alone-

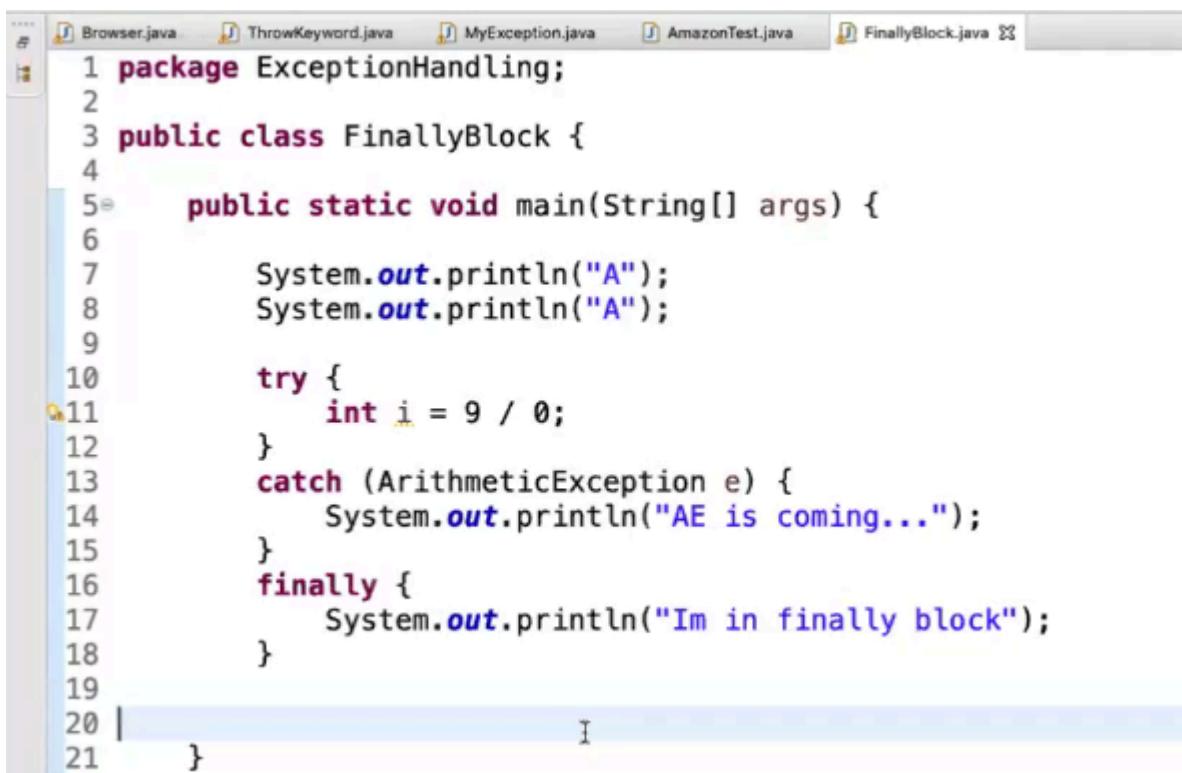
//Syntax error on token "finally", delete this token



```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    }
11
12    finally {
13        System.out.println("Im in finally block");
14    }
15

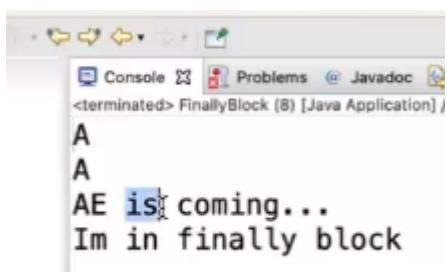
```



```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    try {
11        int i = 9 / 0;
12    }
13    catch (ArithmaticException e) {
14        System.out.println("AE is coming...");
15    }
16    finally {
17        System.out.println("Im in finally block");
18    }
19
20    }
21

```

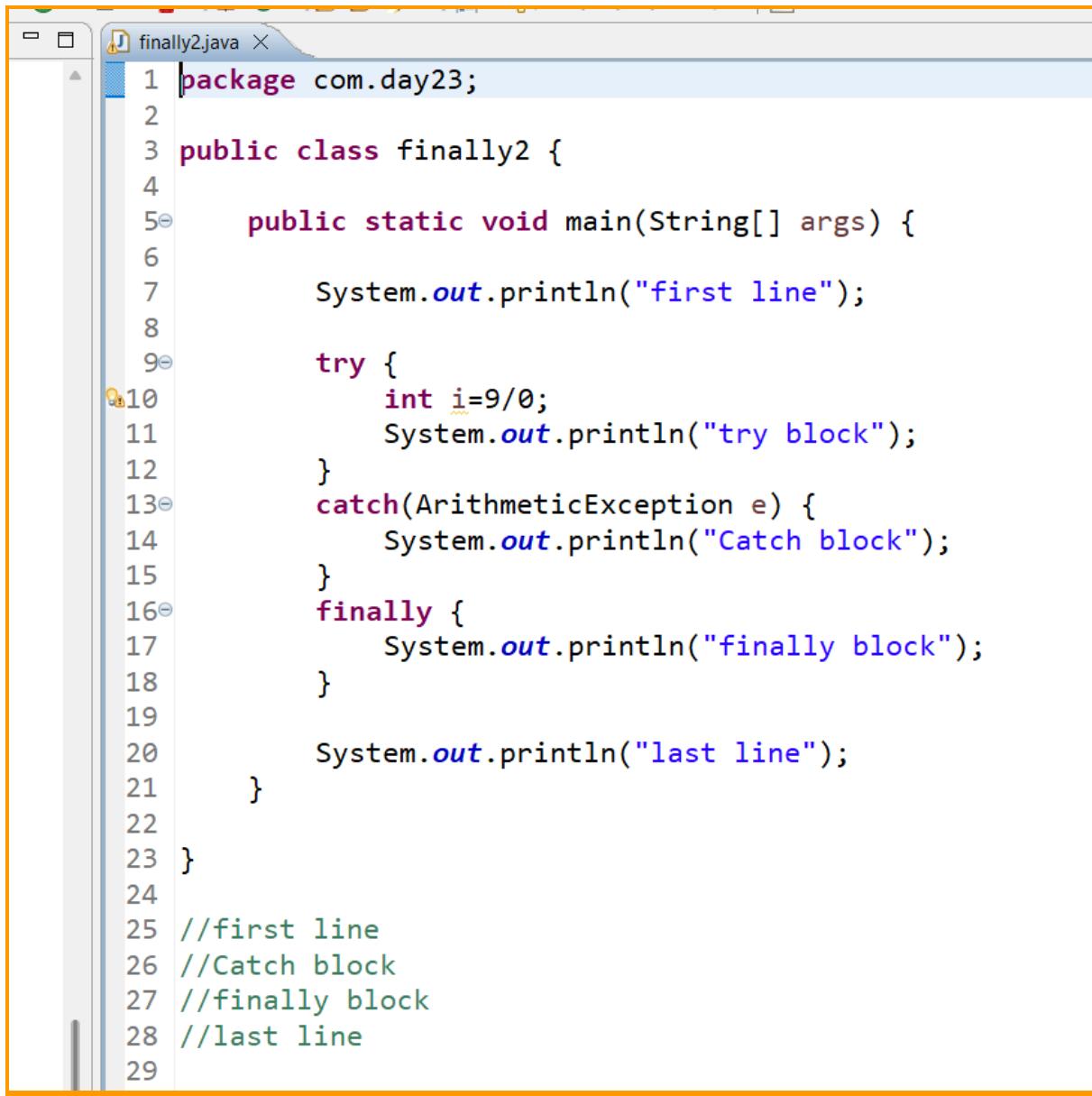


```

Console > Problems > Javadoc >
<terminated> FinallyBlock [8] [Java Application]
A
A
AE is coming...
Im in finally block

```

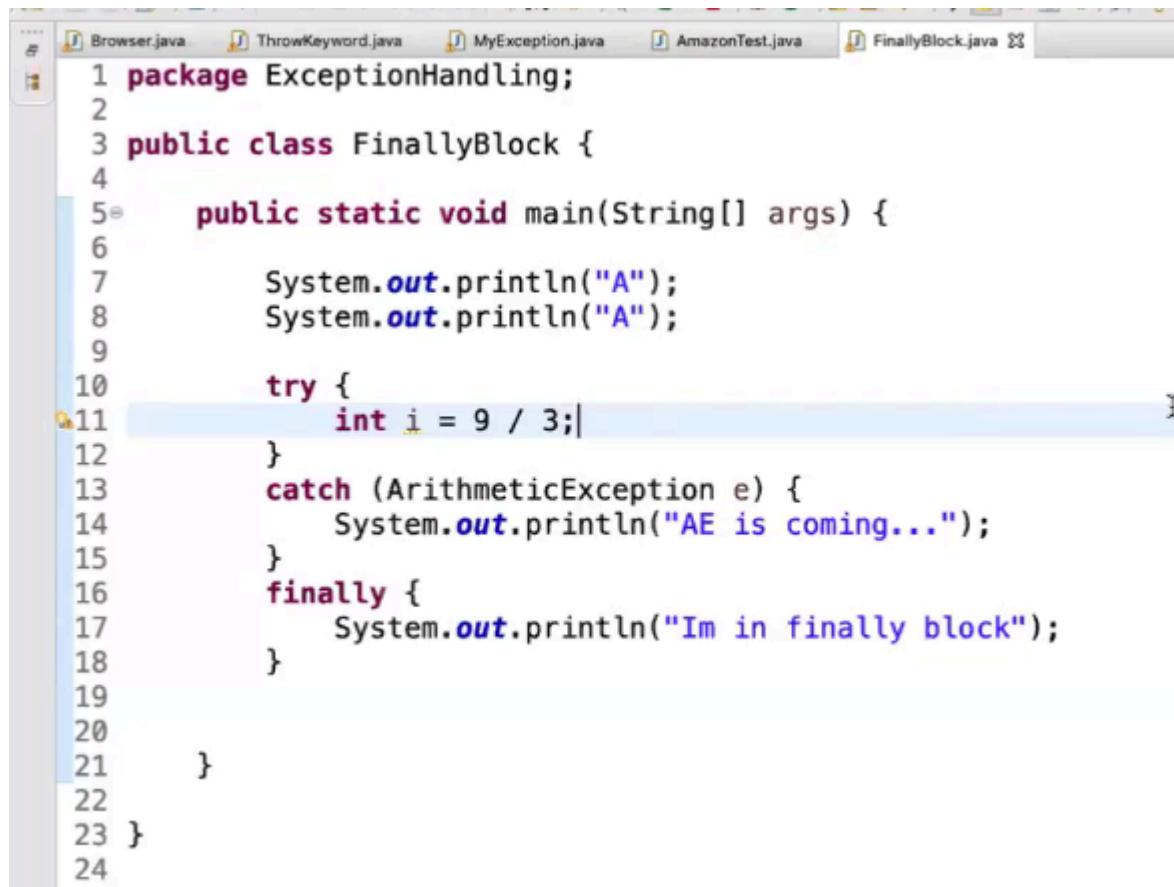
paste finally2-



The screenshot shows a Java code editor window with the file 'finally2.java' open. The code is as follows:

```
1 package com.day23;
2
3 public class finally2 {
4
5     public static void main(String[] args) {
6
7         System.out.println("first line");
8
9         try {
10             int i=9/0;
11             System.out.println("try block");
12         }
13         catch(ArithmaticException e) {
14             System.out.println("Catch block");
15         }
16         finally {
17             System.out.println("finally block");
18         }
19
20         System.out.println("last line");
21     }
22
23 }
24
25 //first line
26 //Catch block
27 //finally block
28 //last line
29
```

No exception code-



```
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 3;
12        }
13        catch (ArithmaticException e) {
14            System.out.println("AE is coming...");
15        }
16        finally {
17            System.out.println("Im in finally block");
18        }
19
20    }
21
22 }
23
24 }
```

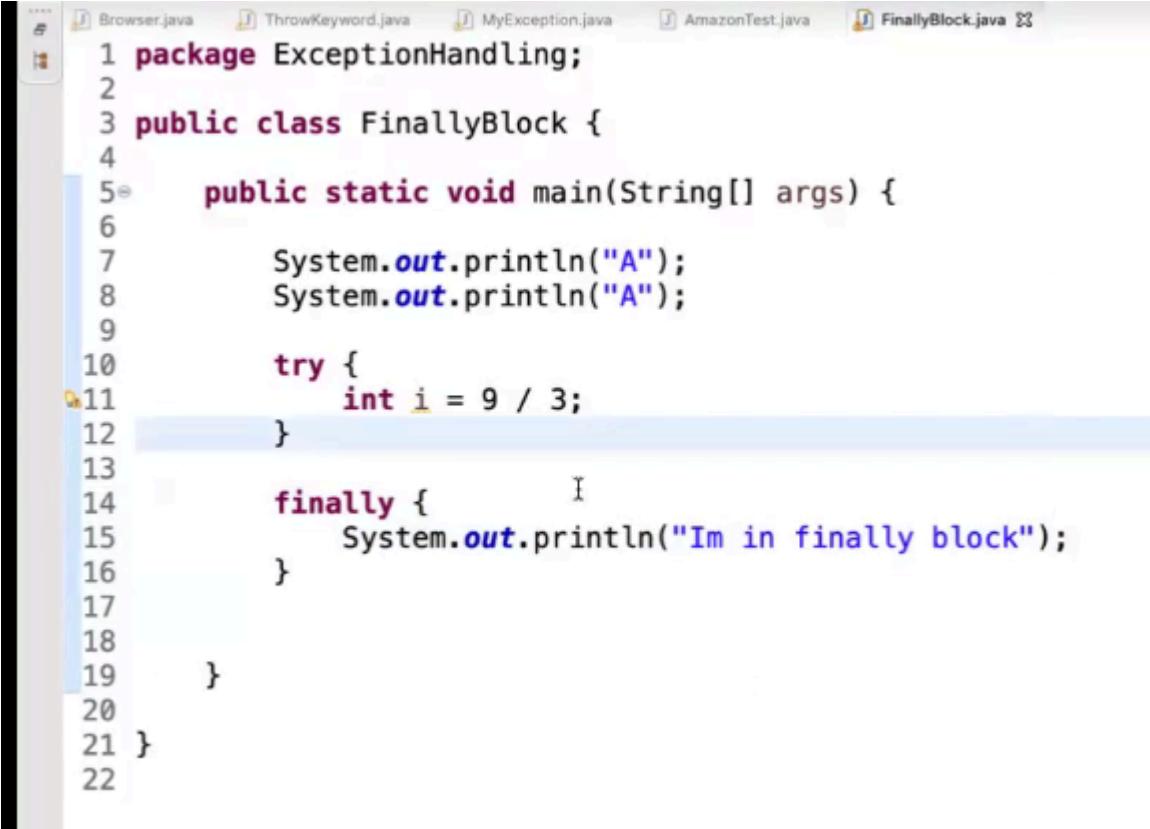


A
A
Im in finally block

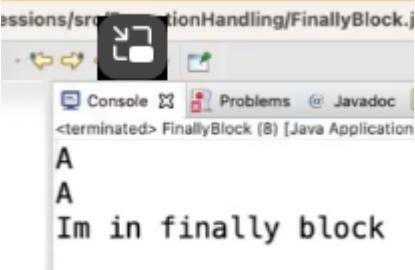
paste finally3-

```
finally3.java
1 package com.day23;
2
3 public class finally3 {
4
5     public static void main(String[] args) {
6
7         System.out.println("first line");
8
9         try {
10             int i=9/3;
11             System.out.println("try block");
12         }
13         catch(ArithmaticException e) {
14             System.out.println("Catch block");
15         }
16         finally {
17             System.out.println("finally block");
18         }
19
20         System.out.println("last line");
21     }
22
23 }
24
25 //first line
26 //try block
27 //finally block
28 //last line
29
30
```

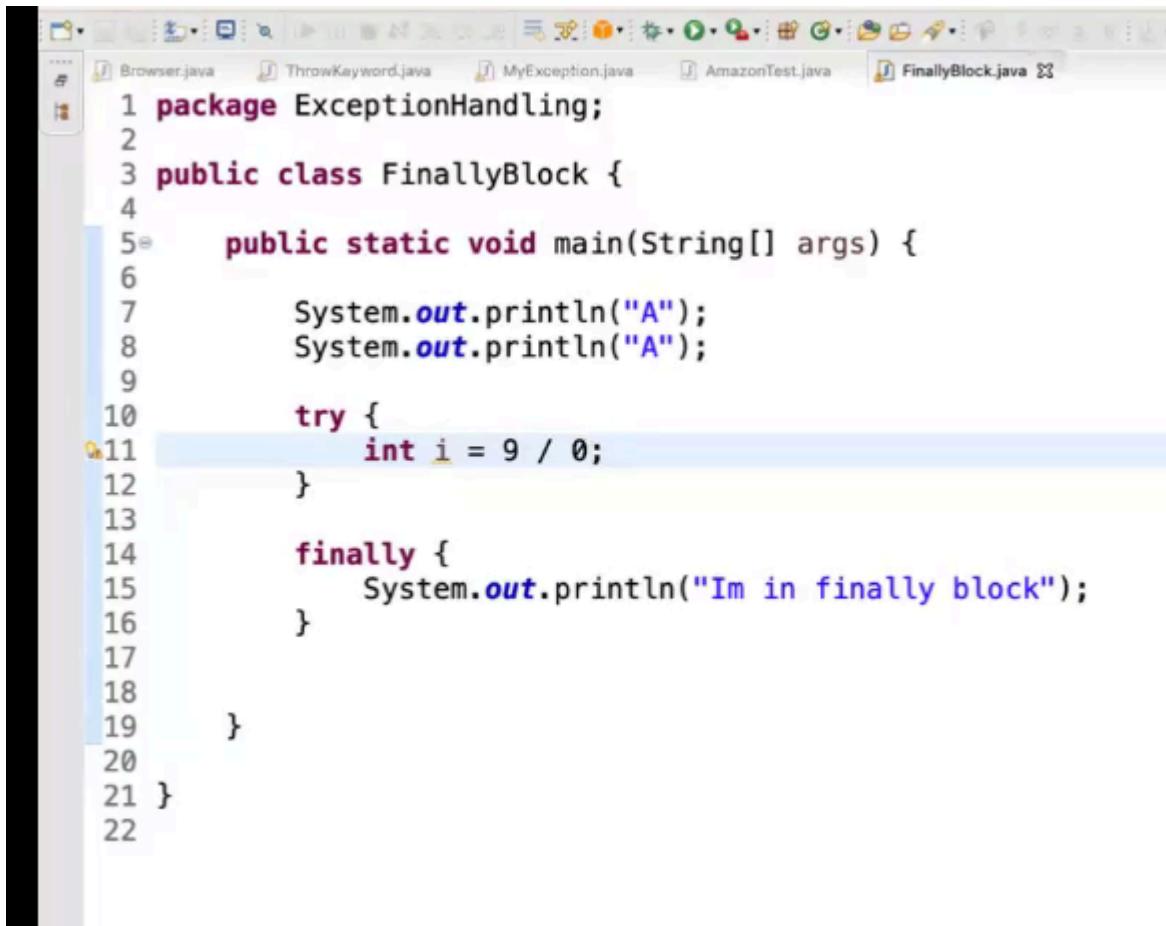
Try finally combination is possible-



```
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    try {
11        int i = 9 / 3;
12    }
13
14    finally {
15        System.out.println("Im in finally block");
16    }
17
18}
19
20}
21
22
```



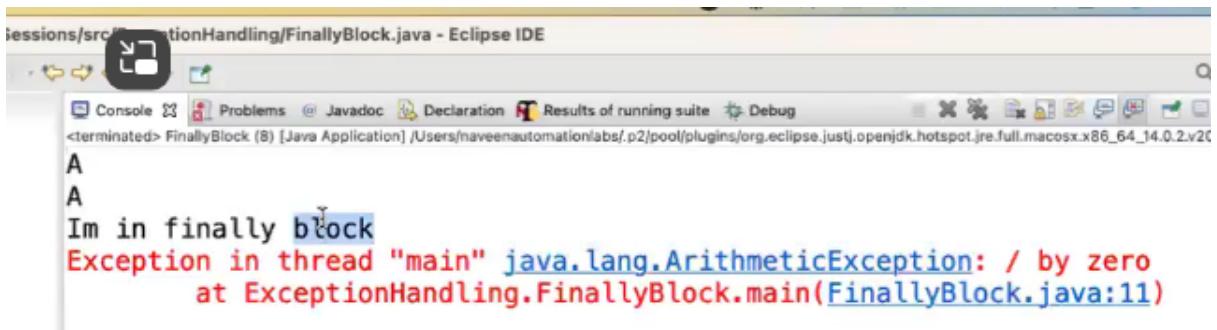
A
A
Im in finally block



```

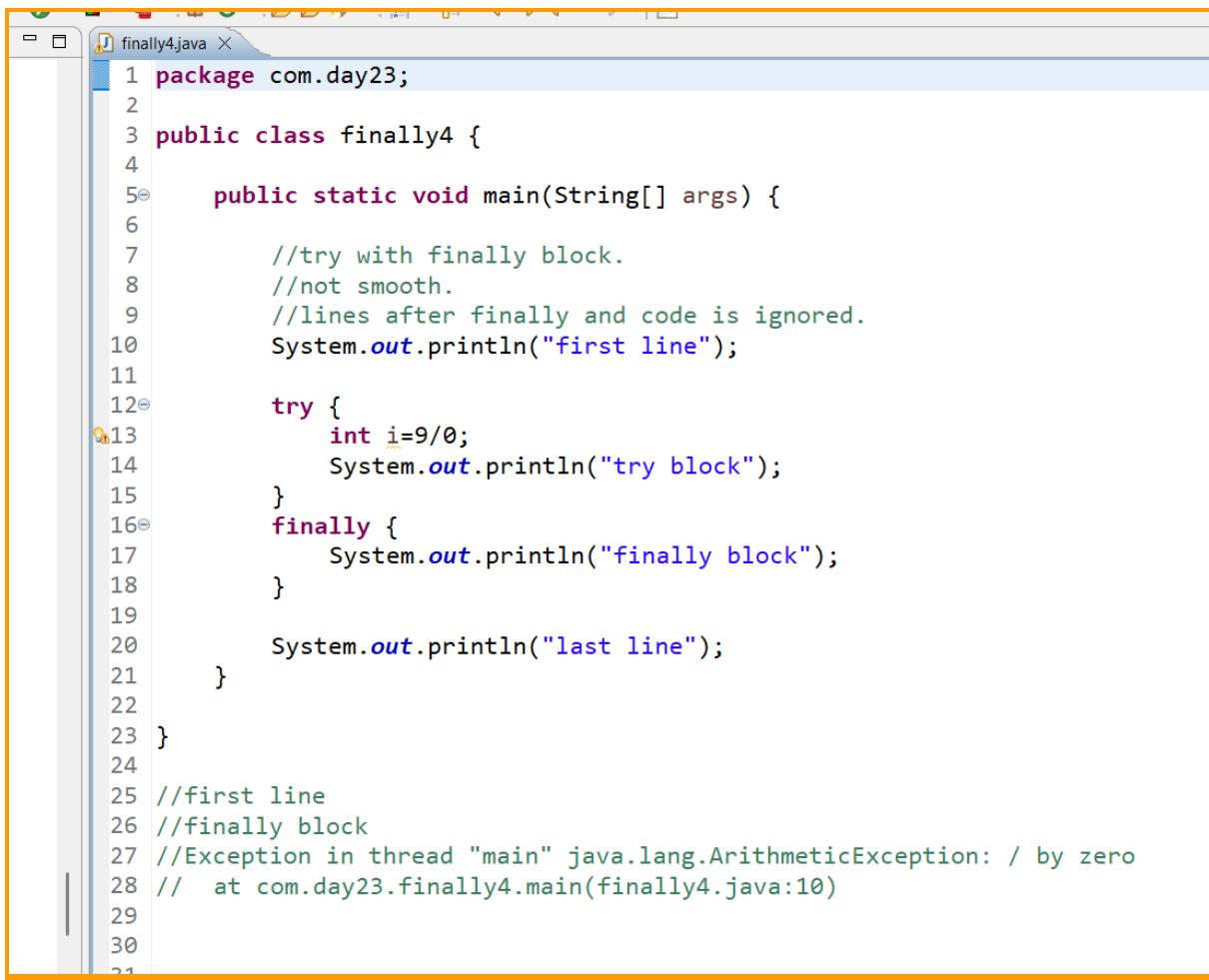
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 0;
12        }
13
14        finally {
15            System.out.println("Im in finally block");
16        }
17
18    }
19
20}
21
22

```



A
A
Im in finally block
Exception in thread "main" java.lang.ArithmaticException: / by zero
at ExceptionHandling.FinallyBlock.main(FinallyBlock.java:11)

paste finally4-

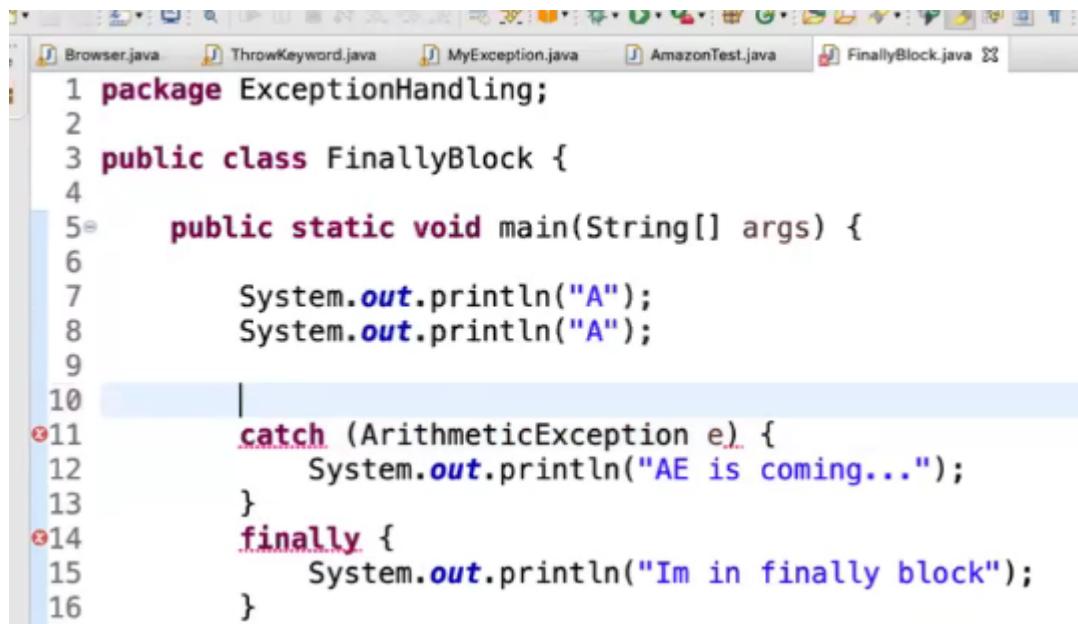


```
1 package com.day23;
2
3 public class finally4 {
4
5     public static void main(String[] args) {
6
7         //try with finally block.
8         //not smooth.
9         //lines after finally and code is ignored.
10        System.out.println("first line");
11
12        try {
13            int i=9/0;
14            System.out.println("try block");
15        }
16        finally {
17            System.out.println("finally block");
18        }
19
20        System.out.println("last line");
21    }
22
23 }
24
25 //first line
26 //finally block
27 //Exception in thread "main" java.lang.ArithmetricException: / by zero
28 // at com.day23.finally4.main(finally4.java:10)
29
30
31
```

Catch and finally cannot come without try-

//Syntax error on token "catch", (expected

//Syntax error on token "finally", delete this token

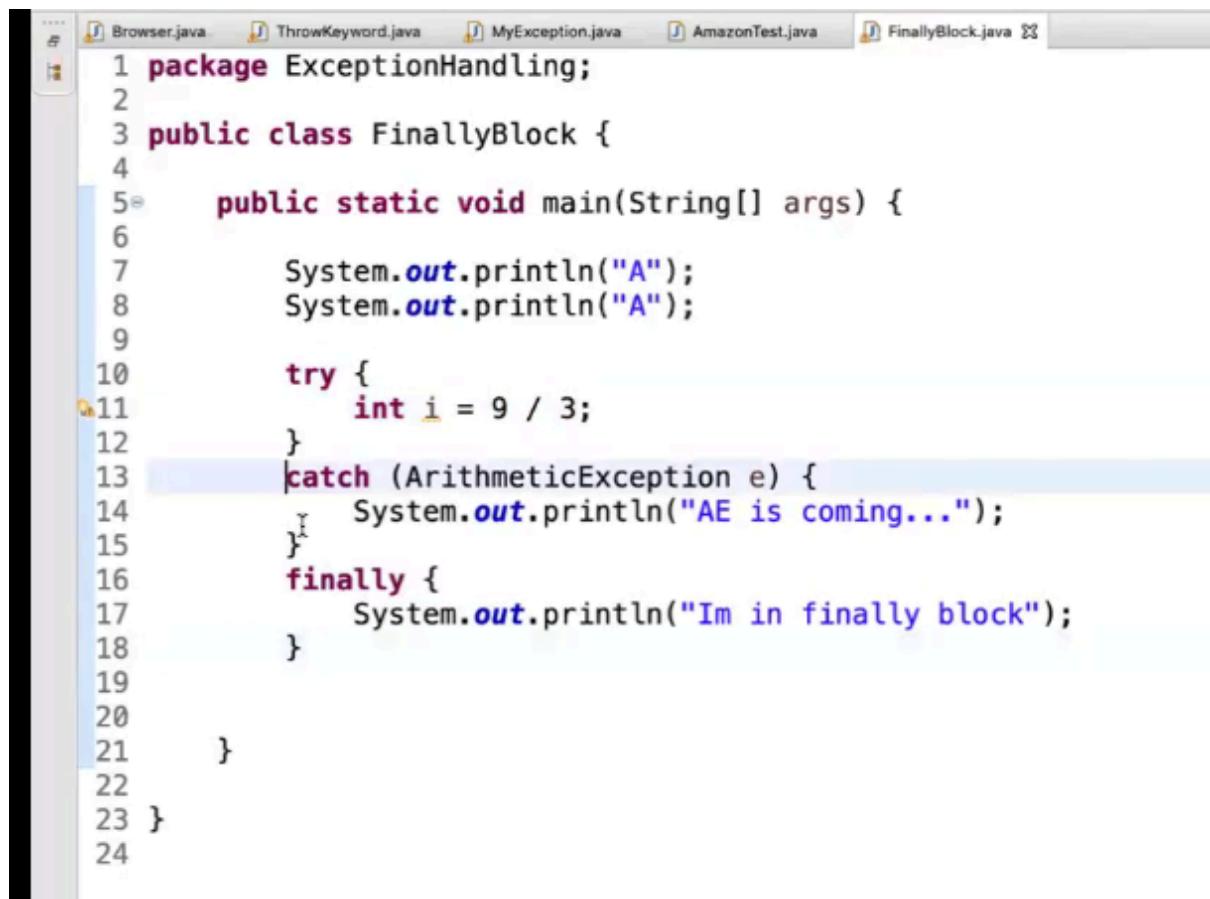


```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    }
11
12    catch (ArithmaticException e) {
13        System.out.println("AE is coming...");
14    }
15
16    finally {
17        System.out.println("Im in finally block");
18    }
19
20
21 }
22
23 }
24

```

Best combination-



```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    try {
11        int i = 9 / 3;
12    }
13    catch (ArithmaticException e) {
14        System.out.println("AE is coming...");
15    }
16    finally {
17        System.out.println("Im in finally block");
18    }
19
20
21 }
22
23 }
24

```

Multiple finally not allowed-

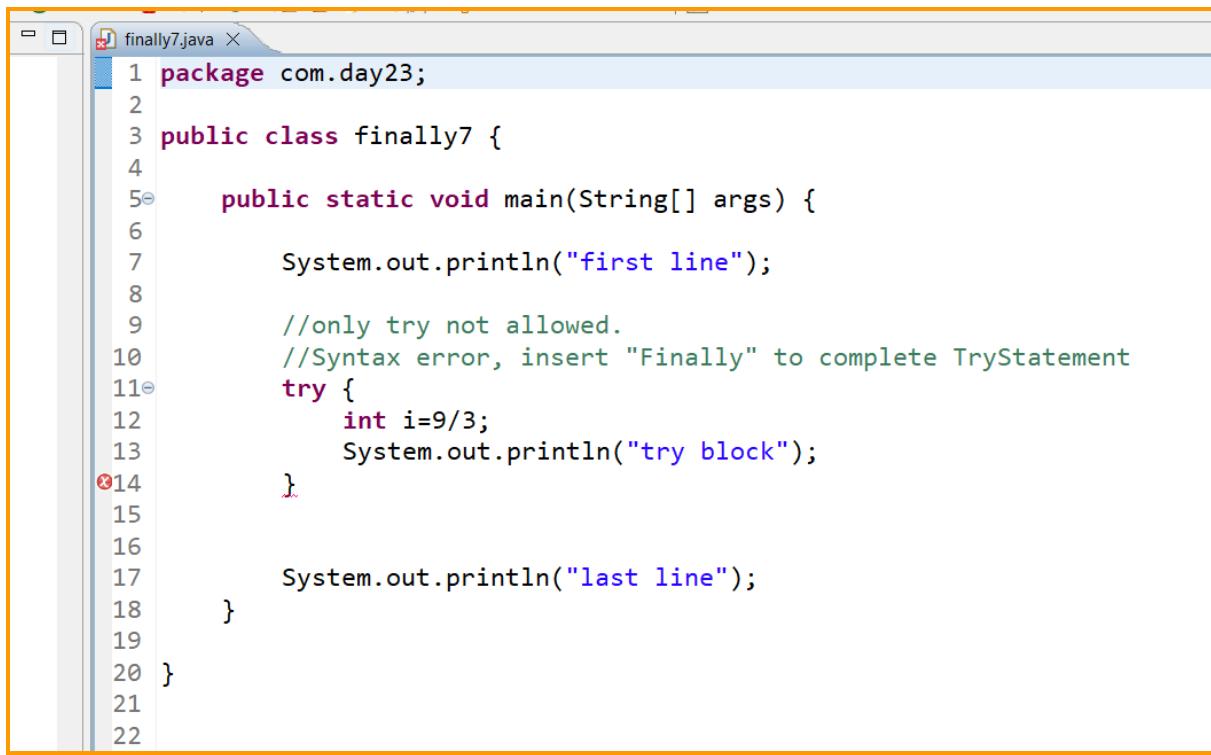
//Syntax error on token "finally", delete this token

```

6
7     System.out.println("A");
8     System.out.println("A");
9
10    try {
11        int i = 9 / 3;
12    }
13    catch (ArithmaticException e) {
14        System.out.println("AE is coming...");
15    }
16    finally {
17        System.out.println("Im in finally block");
18    }
19    finally {
20        System.out.println("Im in finally block");
21    }
22
23}
24
25

```

paste finally7-



The screenshot shows a Java code editor with a file named "finally7.java". The code contains several syntax errors:

```

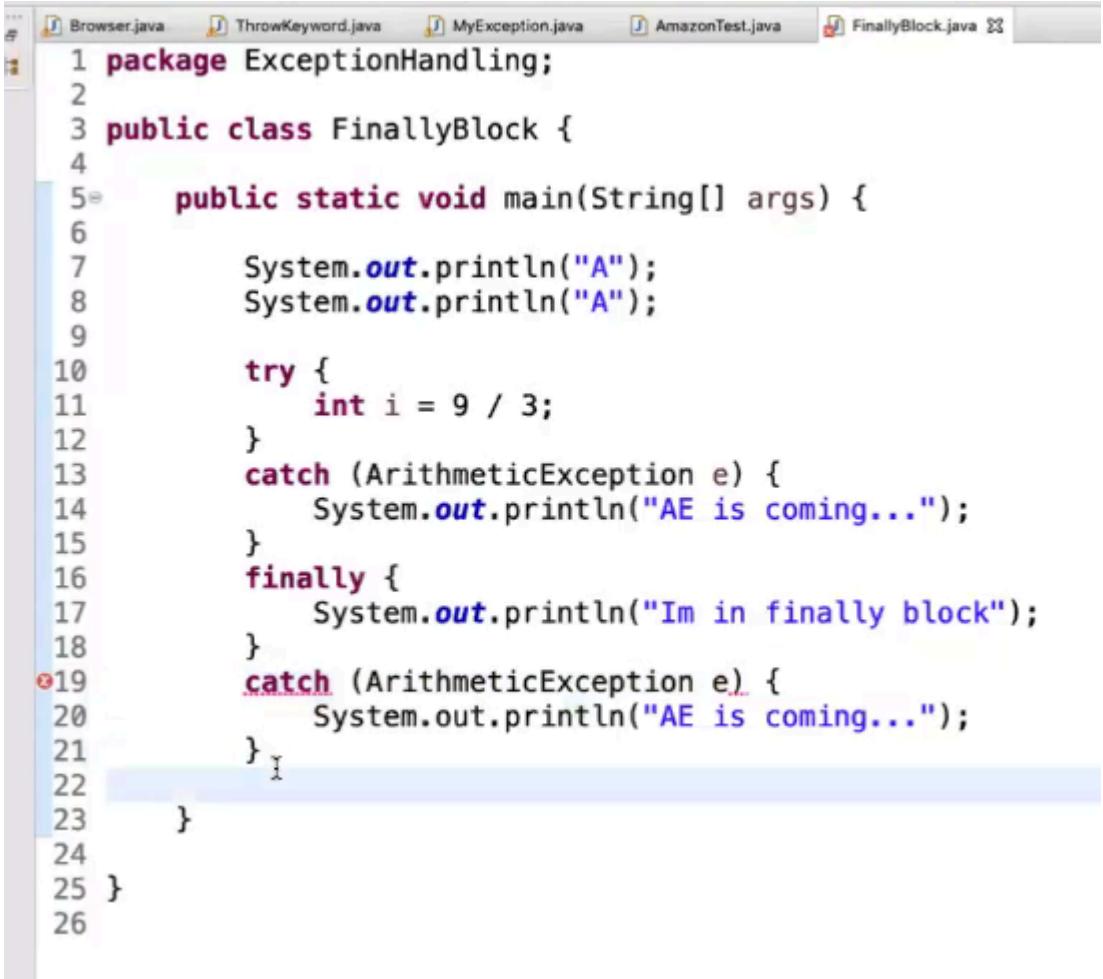
1 package com.day23;
2
3 public class finally7 {
4
5     public static void main(String[] args) {
6
7         System.out.println("first line");
8
9         //only try not allowed.
10        //Syntax error, insert "Finally" to complete TryStatement
11        try {
12            int i=9/3;
13            System.out.println("try block");
14        }
15
16        System.out.println("last line");
17    }
18}
19
20
21
22

```

The errors are indicated by red squiggly lines under the "try" keyword at line 11 and the closing brace at line 14. The code is intended to demonstrate an invalid sequence of statements within a try block.

Improper sequence not allowed-

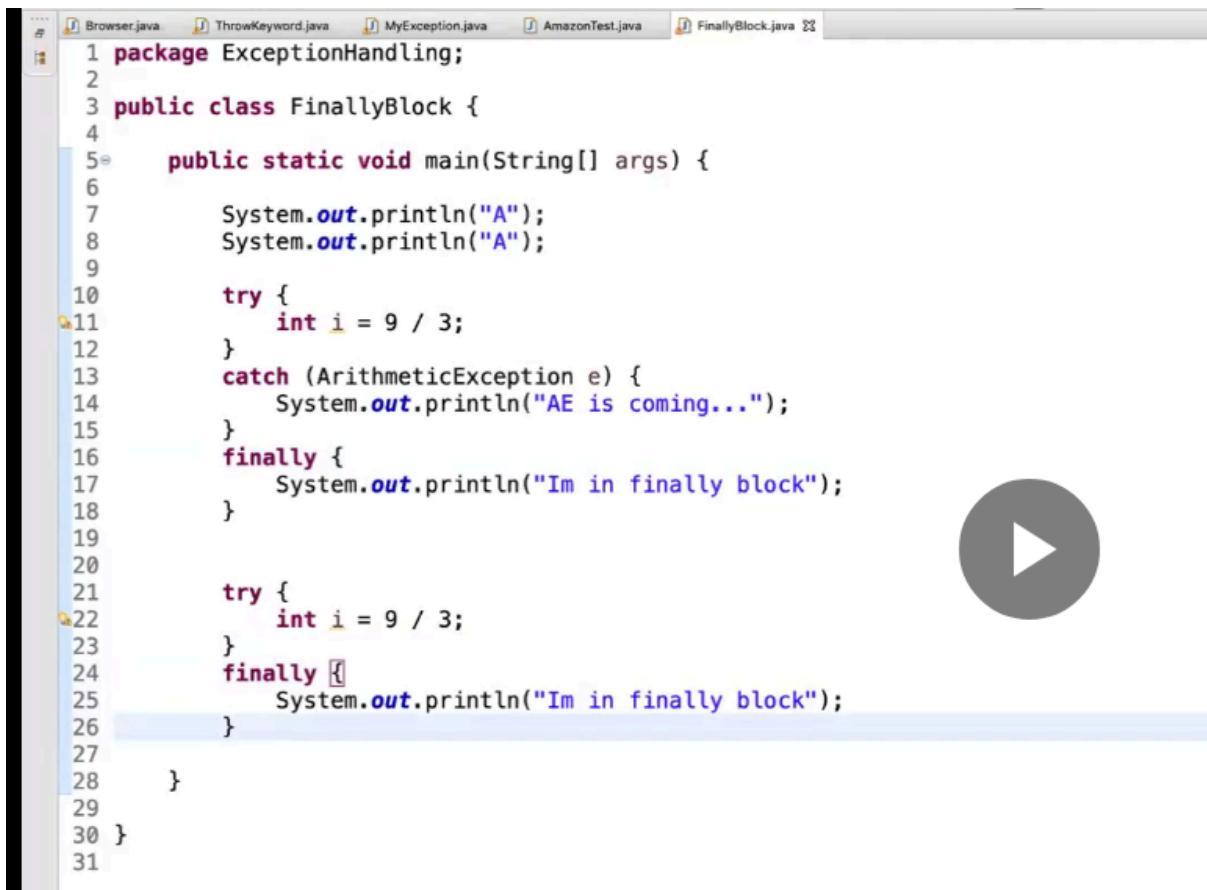
//Syntax error on token "catch", invalid RecordHeaderName



The screenshot shows a Java code editor with the file 'FinallyBlock.java' open. The code demonstrates nested try-catch blocks and finally blocks. It prints 'A' twice, then attempts to divide by zero, which triggers an ArithmeticException. Both catch blocks print 'AE is coming...', and the finally block prints 'Im in finally block'. The code is as follows:

```
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 3;
12        }
13        catch (ArithmaticException e) {
14            System.out.println("AE is coming...");
15        }
16        finally {
17            System.out.println("Im in finally block");
18        }
19        catch (ArithmaticException e) {
20            System.out.println("AE is coming...");
21        }
22    }
23
24
25 }
26
```

Multiply try catch in one class possible-



```
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 3;
12        } catch (ArithmaticException e) {
13            System.out.println("AE is coming...");
14        } finally {
15            System.out.println("Im in finally block");
16        }
17
18        try {
19            int i = 9 / 3;
20        } finally {
21            System.out.println("Im in finally block");
22        }
23    }
24
25 }
```

paste finally9-

```
1 package com.day23;
2
3 public class finally9 {
4
5     public static void main(String[] args) {
6
7         System.out.println("first line");
8
9         try {
10             int i=9/0;
11             System.out.println("try block");
12         }
13         catch(ArithmetricException e) {
14             System.out.println("Catch block");
15         }
16         finally {
17             System.out.println("finally block");
18         }
19
20
21         try {
22             String s=null;
23             s="karan";
24             System.out.println(s.length());
25         }
26     }
27 }
```

```
24     System.out.println("last line");
25 }
26 }
27 } catch(NullPointerException e) {
28     System.out.println("Catch block");
29 }
30 } finally {
31     System.out.println("finally block");
32 }
33 System.out.println("last line");
34 }
35
36 }
37
38 //first line
39 //Catch block
40 //finally block
41 //5
42 //finally block
43 //last line
44
45
46
47
```

paste finally10-

The screenshot shows a Java code editor window with the file 'finally10.java' open. The code demonstrates exception handling with nested try-catch blocks and finally blocks.

```
1 package com.day23;
2
3 public class finally10 {
4
5     public static void main(String[] args) {
6
7         System.out.println("first line");
8
9         try {
10            int i=9/0;
11            System.out.println("try block");
12        }
13        catch(ArithmaticException e) {
14            System.out.println("Catch block");
15        }
16        finally {
17            System.out.println("finally block");
18        }
19
20
21        try {
22            String s=null;
23            System.out.println(s.length());
24
25        }
26        catch(NullPointerException e) {
27            System.out.println("Catch block");
28        }
29    }
}
```

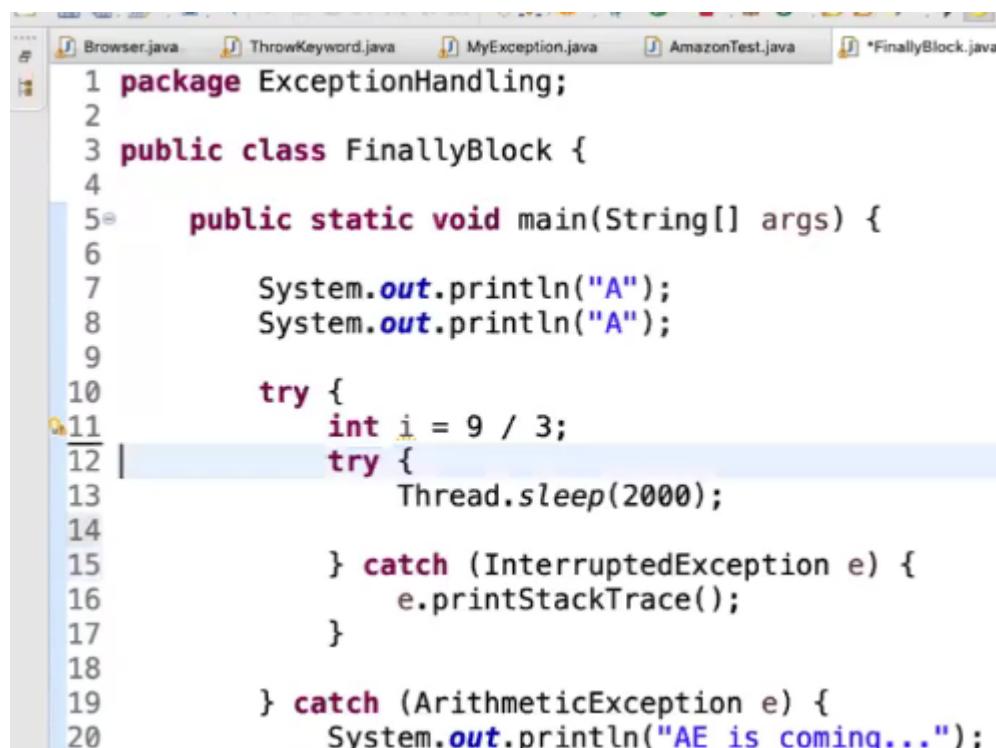
```

27         System.out.println("Catch block");
28     }
29     finally {
30         System.out.println("finally block");
31     }
32     System.out.println("last line");
33 }
34
35 }
36
37 //first line
38 //Catch block
39 //finally block
40 //Catch block
41 //finally block
42 //last line
43

```

Any damn code can be present inside try catch finally-

Nesting also possible.



The screenshot shows a Java code editor with several tabs at the top: Browser.java, ThrowKeyword.java, MyException.java, AmazonTest.java, and *FinallyBlock.java. The code in *FinallyBlock.java is as follows:

```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 3;
12            try {
13                Thread.sleep(2000);
14
15            } catch (InterruptedException e) {
16                e.printStackTrace();
17            }
18
19        } catch (ArithmaticException e) {
20            System.out.println("AE is coming...");
21        }
22    }
23}

```

The line `12 |` is highlighted in blue, indicating the current line of code being edited.

paste finally12-

```

finally12.java X
1 package com.day23;
2
3 public class finally12 {
4
5     public static void main(String[] args) {
6
7 //        catch within catch not allowed.
8
9         System.out.println("first line");
10
11    try {
12        int i=9/0;
13        System.out.println("try block");
14        try {
15            Thread.sleep(2000);
16        }
17        catch(ArithmeticException e) {
18            System.out.println("catch block");
19        }
20        finally {
21            System.out.println("inner block");
22        }
23    }
24    catch(ArithmeticException e) {
25        System.out.println("Catch block");//Syntax error, insert "}" to complete Block
26
27    catch(ArithmeticException e) {
28        System.out.println("catch block");
29    }
30 } finally {
31
32     System.out.println("catch block");
33 }
34
35
36    try {
37        String s=null;
38        System.out.println(s.length());
39    }
40    catch(NullPointerException e) {
41        System.out.println("Catch block");
42    }
43    finally {
44        System.out.println("finally block");
45        //Syntax error on token "finally", delete this token
46        //finally within finally not allowed.
47        finally {
48            System.out.println("finally block");
49        }
50
}

```

```
49             System.out.println("finally block");
50         }
51     }
52     System.out.println("last line");
53 }
54
55 }
56
57
58
59
60 }
```



```
1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    try {
11        int i = 9 / 3;
12
13    }
14
15
16 //    catch (ArithmaticException e) {
17 //        System.out.println("AE is coming...");
18 //    }
19
20    finally {
21        System.out.println("Im in finally block");
22    }
23
24
25    System.out.println("bye");
26
27 }
28
29 }
30 }
```

iavaSessions/src/ExceptionHandling/FinallyBlock.java - E

```

A
A
Im in finally block
ring[ bye

```

Browser.java ThrowKeyword.java MyException.java AmazonTest.java FinallyBlock.java

```

1 package ExceptionHandling;
2
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10    try {
11        int i = 9 / 0;
12
13    }
14
15
16 //    catch (ArithmaticException e) {
17 //        System.out.println("AE is coming....");
18 //    }
19
20    finally {
21        System.out.println("Im in finally block");
22    }
23
24
25    System.out.println("bye");
26
27 }
28
29 }
30

```

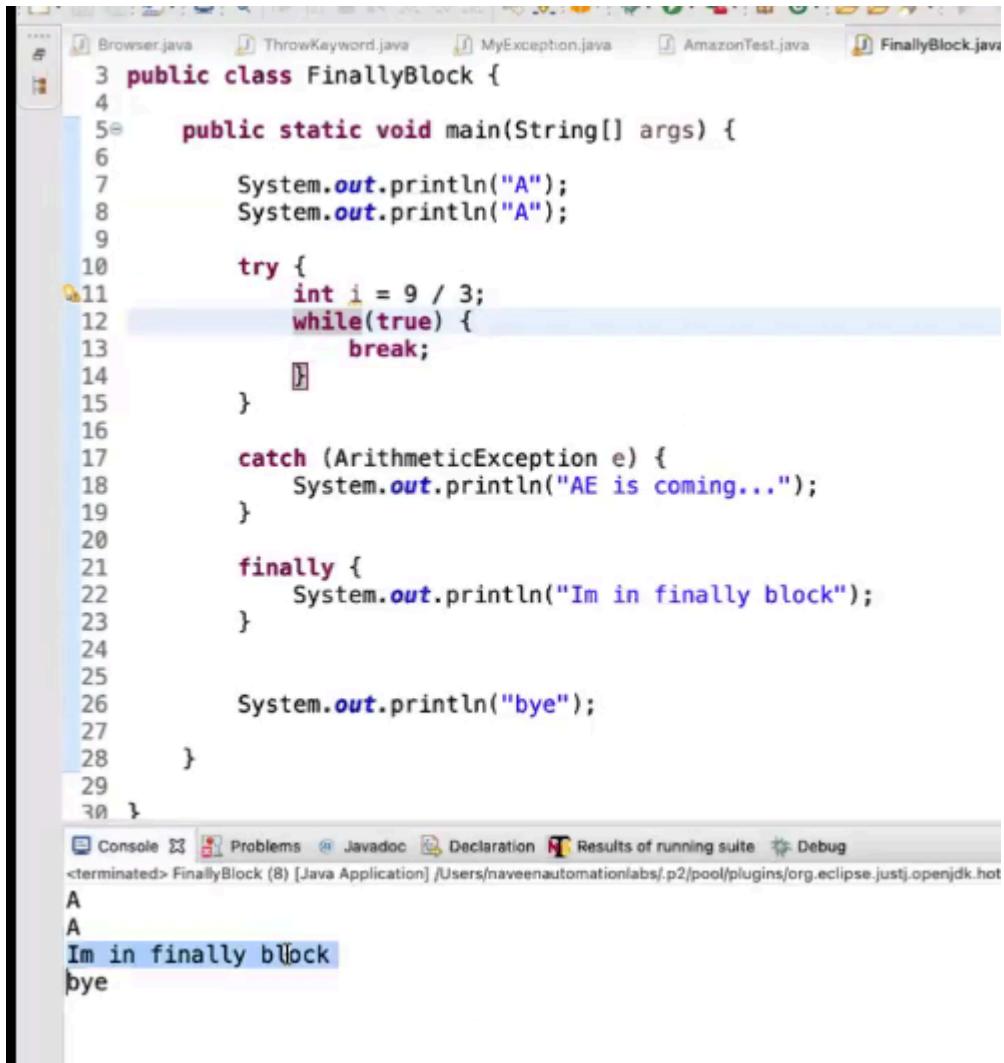
ssions/src/ExceptionHandling/FinallyBlock.java - Eclipse IDE

```

A
A
Im in finally block
Exception in thread "main" java.lang.ArithmaticException: / by zero
at ExceptionHandling.FinallyBlock.main(FinallyBlock.java:11)

```

With loops inside try-



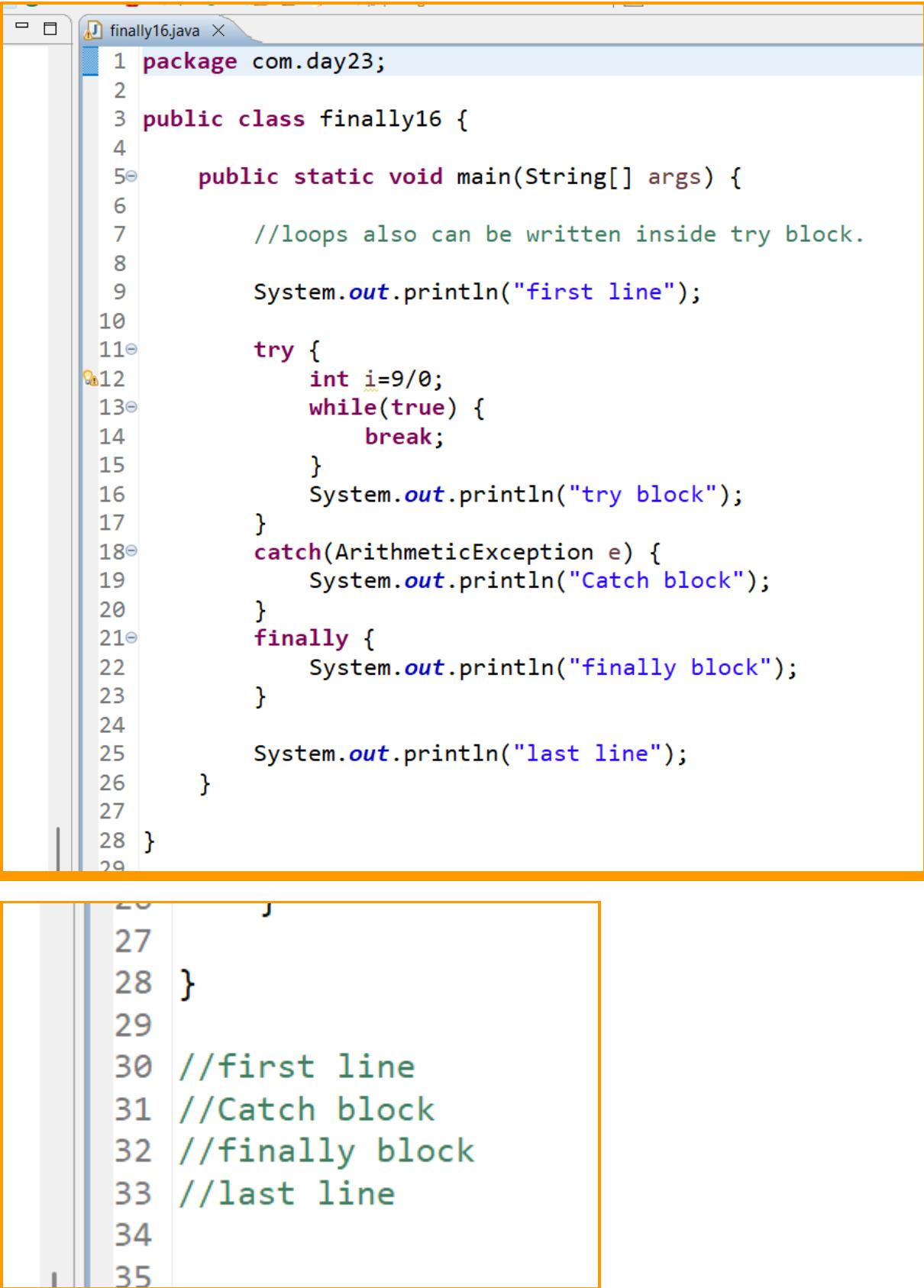
The screenshot shows the Eclipse IDE interface with the code editor open to a Java file named FinallyBlock.java. The code contains a main method with a try block that includes a while loop. The finally block is executed even when an exception is thrown from the try block. The output window shows the console output with the expected results.

```
3 public class FinallyBlock {
4
5     public static void main(String[] args) {
6
7         System.out.println("A");
8         System.out.println("A");
9
10        try {
11            int i = 9 / 3;
12            while(true) {
13                break;
14            }
15        }
16
17        catch (ArithmaticException e) {
18            System.out.println("AE is coming...");
19        }
20
21        finally {
22            System.out.println("Im in finally block");
23        }
24
25
26        System.out.println("bye");
27
28    }
29
30 }
```

Console Output:

```
A
A
Im in finally block
bye
```

paste finally16-



```
finally16.java
1 package com.day23;
2
3 public class finally16 {
4
5     public static void main(String[] args) {
6
7         //loops also can be written inside try block.
8
9         System.out.println("first line");
10
11        try {
12            int i=9/0;
13            while(true) {
14                break;
15            }
16            System.out.println("try block");
17        }
18        catch(ArithmaticException e) {
19            System.out.println("Catch block");
20        }
21        finally {
22            System.out.println("finally block");
23        }
24
25        System.out.println("last line");
26    }
27
28 }
29
30 //first line
31 //Catch block
32 //finally block
33 //last line
34
35
```

paste finally17-

```
finally17.java
1 package com.day23;
2
3 public class finally17 {
4
5     public static void main(String[] args) {
6
7         //loops also can be written inside try block.
8
9         System.out.println("first line");
10
11        try {
12            int i=9/0;
13            while(true) {
14                break;
15            }
16            System.out.println("try block");
17        }
18        catch(ArithmetricException e) {
19            System.out.println("Catch block");
20            for(int i=1;i<=10;i++) {
21                System.out.println(i);
22            }
23        }
24        finally {
25            System.out.println("finally block");
26            for(int i=21;i<=30;i++) {
27                System.out.println(i);
28            }
29        }
30    }
}
```

```
28         }
29     }
30
31     System.out.println("last line");
32 }
33
34 }
35
36 //first line
37 //Catch block
38 //1
39 //2
40 //3
41 //4
42 //5
43 //6
44 //7
45 //8
46 //9
47 //10
48 //finally block
49 //21
```

```
47 //10
48 //finally block
49 //21
50 //22
51 //23
52 //24
53 //25
54 //26
55 //27
56 //28
57 //29
58 //30
59 //last line
60
61
```

Using throws we can throw multiple exceptions-

```
22 }
23
24 // HDFC bank application: 3rd party system
25 public void bankPayment() throws ArithmeticException, NullPointerException {
26     System.out.println("bankPayment method");
27     int i = 9 / 0;
28 }
```



paste throws1-

```
throws1.java
1 package com.day23;
2
3 public class throws1 {
4
5     //throws multiple exceptions.
6
7     //method 1.
8     public void m1() throws ArithmeticException, IndexOutOfBoundsException{
9         System.out.println("m1 method");
10        m2();
11    }
12
13     //method 2.
14     public void m2() throws ArithmeticException{
15         System.out.println("m2 method");
16        m3();
17    }
18
19     //method 3.
20     public void m3() throws ArithmeticException, NullPointerException{
21         System.out.println("m3 method");
22         int i=9/0;
23     }
24
25     //main method.
26     public static void main(String[] args) {
27
28         throws1 b1=new throws1();
29         b1.m1();
30     }
31
32 }
33
34
35 //m1 method
36 //m2 method
37 //m3 method
38 //Exception in thread "main" java.lang.Arithme
39 //tException: / by zero
40 // at com.day23.throws1.m3(throws1.java:20)
41 // at com.day23.throws1.m2(throws1.java:14)
42 // at com.day23.throws1.m1(throws1.java:8)
43 // at com.day23.throws1.main(throws1.java:27)
44
```

paste throws3-

```

1 package com.day23;
2
3 public class throws3 {
4
5     //method 1.
6     //IOException cannot be resolved to a type
7     public void m1() throws ArithmeticException, IOException{
8         System.out.println("m1 method");
9         m2();
10    }
11
12     //method 2.
13     public void m2() throws IndexOutOfBoundsException{
14         System.out.println("m2 method");
15         m3();
16    }
17
18     //method 3.
19     public void m3() throws ArithmeticException, NullPointerException{
20         System.out.println("m3 method");
21         int i=9/0;
22    }
23
24     //main method.
25     public static void main(String[] args) {
26
27         throws3 b1=new throws3();
28         b1.m1(); //The method m1() from the type throws3 refers to the missing type IOException
29         //when exception wont come of that type.
30
31     }
32
33 }
34
35 //m1 method
36 //m2 method
37 //m3 method
38 //Exception in thread "main" java.lang.ArithmaticException: / by zero
39 //  at com.day23.throws1.m3(throws1.java:20)
40 //  at com.day23.throws1.m2(throws1.java:14)
41 //  at com.day23.throws1.m1(throws1.java:8)
42 //  at com.day23.throws1.main(throws1.java:27)
43
44

```

paste throws4-

```

1 package com.day23;
2
3 import java.sql.SQLDataException;
4
5 public class throws4 {
6
7     //method 1.
8
9     public void m1() throws ArithmeticException, SQLDataException{
10         System.out.println("m1 method");
11         m2();
12     }
13
14     //method 2.
15     public void m2() throws IndexOutOfBoundsException{
16         System.out.println("m2 method");
17         m3();
18     }
19
20     //method 3.
21     public void m3() throws ArithmeticException, NullPointerException{
22         System.out.println("m3 method");
23         int i=9/0;
24     }
25
26     //main method.
27     public static void main(String[] args) {
28
29         //main method.
30         throws4 b1=new throws4();
31         b1.m1(); //Unhandled exception type SQLDataException
32         //now code doesn't know how to handle sql data exception.
33     }
34
35 }

```

paste throws5-

```

throws5.java
1 package com.day23;
2
3 import java.sql.SQLDataException;
4
5 public class throws5 {
6
7     //method 1.
8
9     public void m1() throws ArithmeticException, SQLDataException{
10         System.out.println("m1 method");
11         m2();
12     }
13
14     //method 2.
15     public void m2() throws IndexOutOfBoundsException{
16         System.out.println("m2 method");
17         m3();
18     }
19
20     //method 3.
21     public void m3() throws ArithmeticException, NullPointerException{
22         System.out.println("m3 method");
23         int i=9/0;
24     }
25
26     //main method.
27     public static void main(String[] args) throws SQLDataException, ArithmeticException {
28
29         throws5 b1=new throws5();
30         b1.m1(); //Unhandled exception type SQLDataException
31         //now code doesn't know how to handle sql data exception.
32         //to resolve this even main has to implement the sql and arithmetic.
33     }
34
35
36 }
37
38 //m1 method
39 //m2 method
40 //m3 method
41 //Exception in thread "main" java.lang.ArithmetricException: / by zero
42 // at com.day23.throws5.m3(throws5.java:23)
43 // at com.day23.throws5.m2(throws5.java:17)
44 // at com.day23.throws5.m1(throws5.java:11)
45 // at com.day23.throws5.main(throws5.java:30)
46
47
48

```

paste throws2-

```
throws2.java x throws6.java throws7.java throw1.java

1 package com.day23;
2
3 public class throws2 {
4
5     //method 1.
6     public void m1() throws ArithmeticException, IndexOutOfBoundsException{
7         System.out.println("m1 method");
8         m2(); //Unhandled exception type Throwable
9     }
10
11    //method 2.
12    public void m2() throws Throwable, ArithmeticException{
13        System.out.println("m2 method");
14        m3();
15    }
16
17    //method 3.
18    public void m3() throws Exception, NullPointerException{
19        System.out.println("m3 method");
20        int i=9/0;
21    }
22
23    //main method.
24    public static void main(String[] args) {
25
26        throws2 b1=new throws2();
27        b1.m1();
28
29    }
30
31 }
32
33 //m1 method
34 //m2 method
35 //m3 method
36 //Exception in thread "main" java.lang.Arithme
37 //t at com.day23.throws1.m3(throws1.java:20)
38 // at com.day23.throws1.m2(throws1.java:14)
39 // at com.day23.throws1.m1(throws1.java:8)
40 // at com.day23.throws1.main(throws1.java:27)
41
42
```

paste throws6-

The screenshot shows an IDE interface with two tabs open: 'throws6.java' and 'throws7.java'. The 'throws6.java' tab is active, displaying the following code:

```
1 package com.day23;
2
3 public class throws6 {
4
5     //method 1.
6     public void m1() throws Throwable, ArithmeticException, IndexOutOfBoundsException{
7         System.out.println("m1 method");
8         m2(); //Uncaught exception type Throwable
9         //implement throwable here also.
10        //so if lower methods throws exception,
11        //higher methods or main should throw atleast that exception.
12    }
13
14    //method 2.
15    public void m2() throws Throwable, ArithmeticException{
16        System.out.println("m2 method");
17        m3();
18    }
19
20    //method 3.
21    public void m3() throws Exception, NullPointerException{
22        System.out.println("m3 method");
23        int i=9/0;
24    }
25
26    //main method.
27    public static void main(String[] args) {
28
29        throws6 b1=new throws6();
30        b1.m1(); //Uncaught exception type Throwable
31
32    }
33
34 }
35
```

The 'throws7.java' tab is visible in the background. The code in 'throws6.java' includes annotations for various exception types and demonstrates how exception handling propagates from lower-level methods up to the main method.

paste throws7-

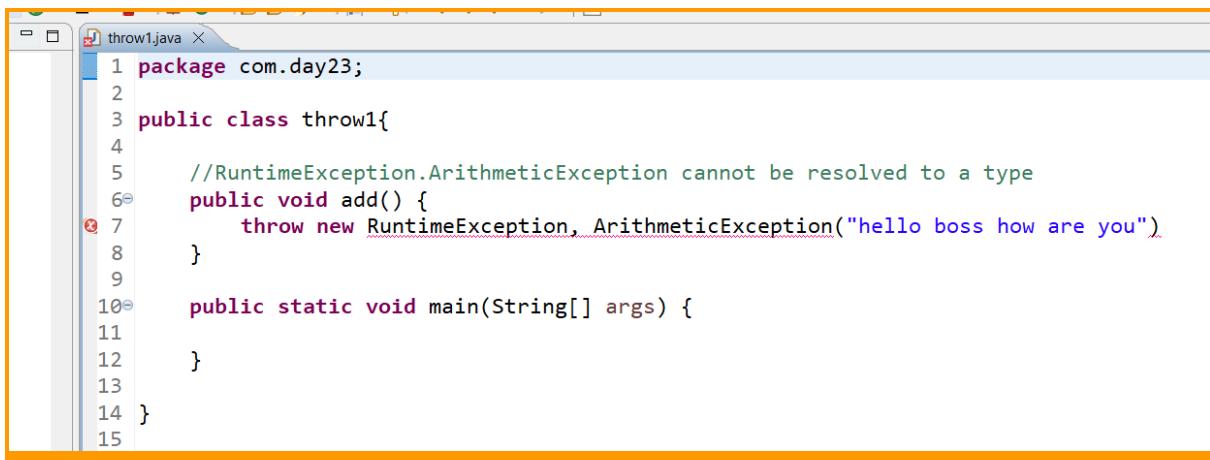
```

1 package com.day23;
2
3 public class throws7 {
4
5     //method 1.
6     public void m1() throws Throwable, ArithmeticException, IndexOutOfBoundsException{
7         System.out.println("m1 method");
8         m2(); //Unhandled exception type Throwable
9         //implement throwable here also.
10        //so if lower methods throws exception,
11        //higher methods or main should throws atleast that exception.
12    }
13
14    //method 2.
15    public void m2() throws Throwable, ArithmeticException{
16        System.out.println("m2 method");
17        m3();
18    }
19
20    //method 3.
21    //method 3 not handling the correct exception
22    //which is arithmetic hence all these issues.
23    public void m3() throws Exception, NullPointerException{
24        System.out.println("m3 method");
25        int i=9/0;
26    }
27
28
29    //main method.
30    public static void main(String[] args) throws
31        ArithmeticException, IndexOutOfBoundsException, Throwable {
32
33        throws7 b1=new throws7();
34        b1.m1(); //Unhandled exception type Throwable
35        //even main method has to throws all exceptions for it to work.
36    }
37
38 }
39
40 //m1 method
41 //m2 method
42 //m3 method
43 //Exception in thread "main" java.lang.ArithmetricException: / by zero
44 // at com.day23.throws7.m3(throws7.java:25)
45 // at com.day23.throws7.m2(throws7.java:17)
46 // at com.day23.throws7.m1(throws7.java:8)
47 // at com.day23.throws7.main(throws7.java:33)
48
49
50
51

```

Using throw only one exception.

paste throw1-



```
1 package com.day23;
2
3 public class throw1{
4
5     //RuntimeException.ArithmetricException cannot be resolved to a type
6     public void add() {
7         throw new RuntimeException, ArithmetricException("hello boss how are you");
8     }
9
10    public static void main(String[] args) {
11
12    }
13
14 }
15
```