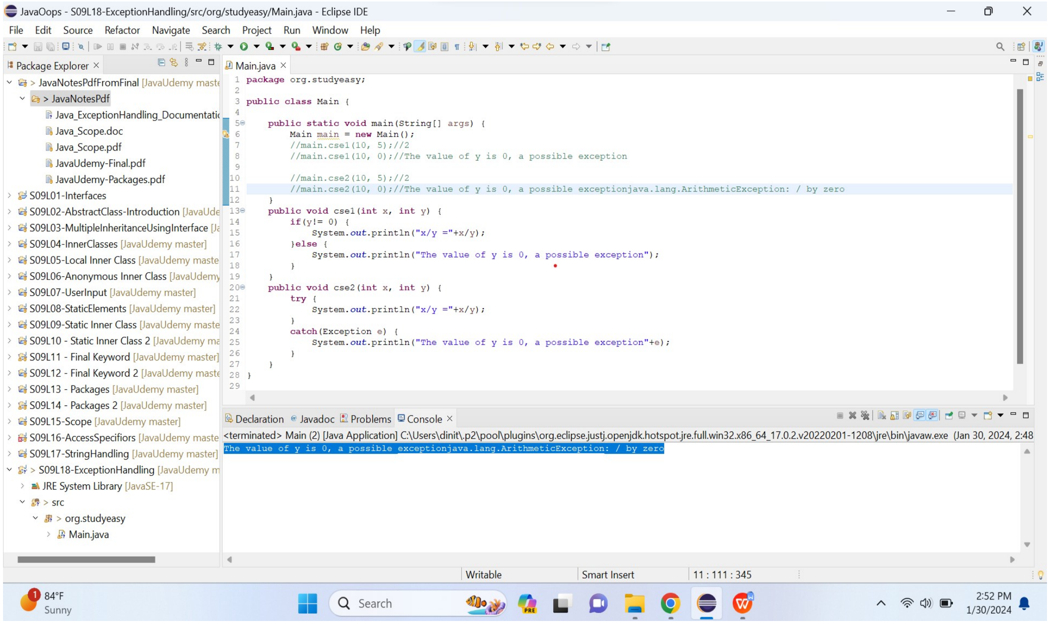
EXCEPTION HANDLING

- **Exception** is the mother of all exceptions.

- Documentation URL - https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/java/lang/Exception.html

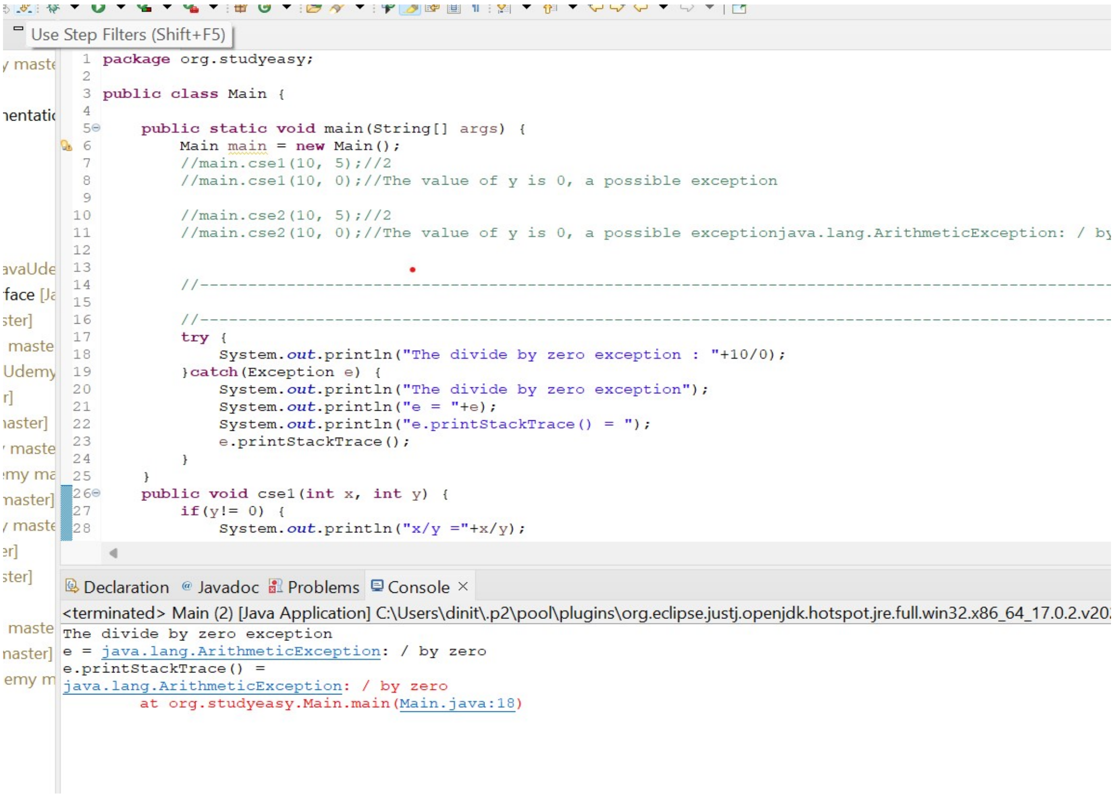
- Below is the example of Java Exception Handling: A difference between Manual and Exception Handling


JavaOops - S09L18-ExceptionHandling/src/org/studyeasy/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Main.java
Declaration Javadoc Problems Console
Writable
Smart Insert
11 : 111 : 345

```java
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        Main main = new Main();
        //main.cse1(10, 5);//2
        //main.cse1(10, 0);//The value of y is 0, a possible exception

        //main.cse2(10, 5);//2
        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero
    }
    public void cse1(int x, int y) {
        if(y!= 0) {
            System.out.println("x/y ="+x/y);
        }else {
            System.out.println("The value of y is 0, a possible exception");
        }
    }
    public void cse2(int x, int y) {
        try {
            System.out.println("x/y ="+x/y);
        }
        catch(Exception e) {
            System.out.println("The value of y is 0, a possible exception"+e);
        }
    }
}
```

```
<terminated> Main (2) [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 30, 2024, 2:48
The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero
```

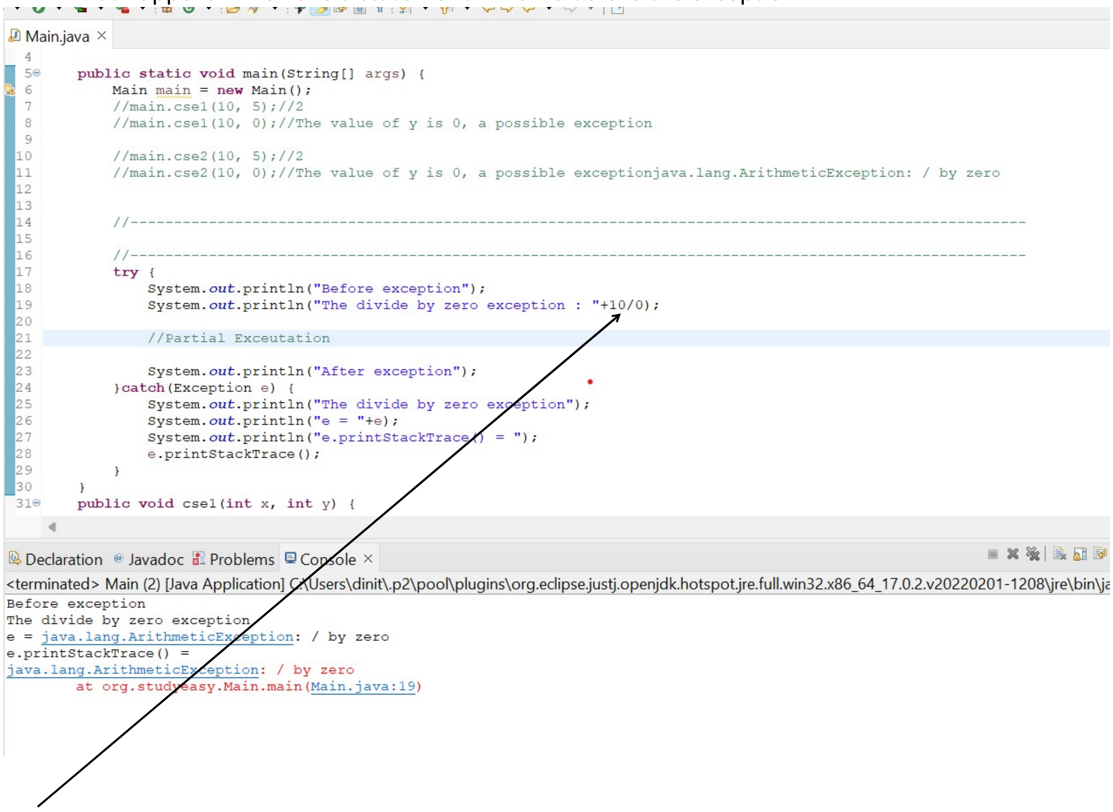
- Method - `e.printStackTrace()` – Used to error handle like also shows on which the line error is.



```
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        Main main = new Main();
        //main.cse1(10, 5);//2
        //main.cse1(10, 0);//The value of y is 0, a possible exception

        //main.cse2(10, 5);//2
        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by


        //-------------------------------------------------------------------------------------------

        //-------------------------------------------------------------------------------------------
        try {
            System.out.println("The divide by zero exception : "+10/0);
        }catch(Exception e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
    }
    public void cse1(int x, int y) {
        if(y!= 0) {
            System.out.println("x/y ="+x/y);
```

```
<terminated> Main (2) [Java Application] C:\Users\dinit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20
The divide by zero exception
e = java.lang.ArithmeticException: / by zero
e.printStackTrace() =
java.lang.ArithmeticException: / by zero
	at org.studyeasy.Main.main(Main.java:18)
```

- What happens if we have a statement which is before the exception?



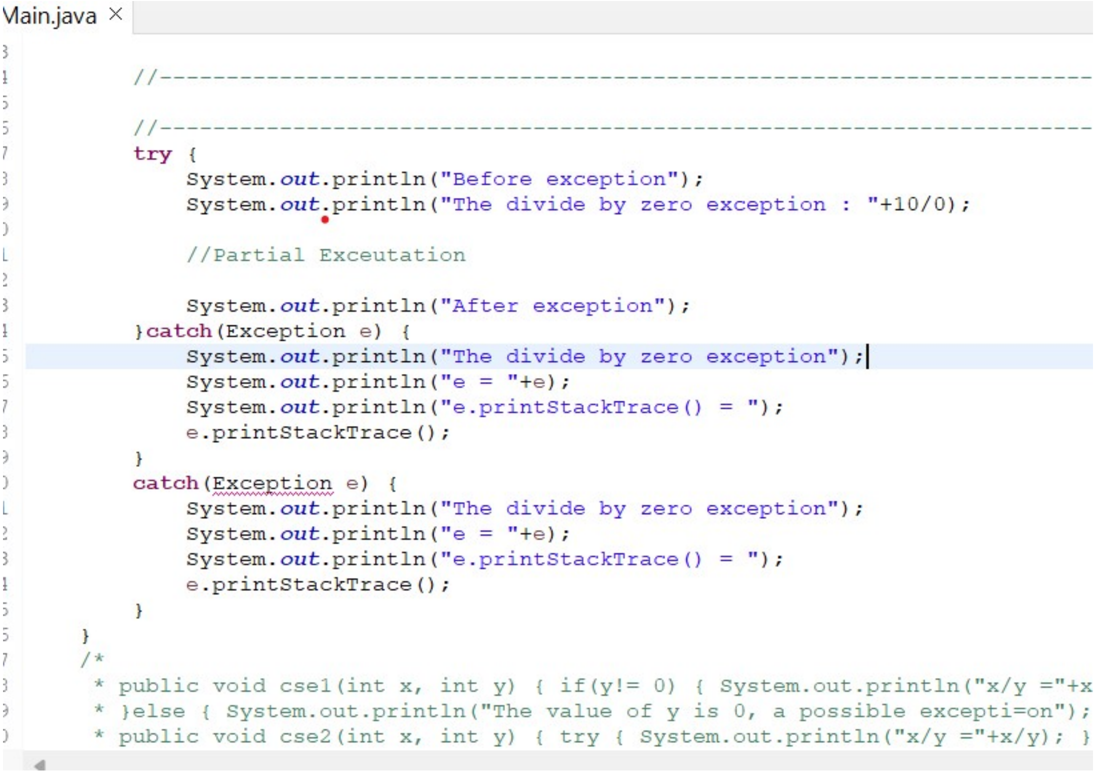
```

    public static void main(String[] args) {
        Main main = new Main();
        //main.cse1(10, 5);//2
        //main.cse1(10, 0);//The value of y is 0, a possible exception

        //main.cse2(10, 5);//2
        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero


        //-------------------------------------------------------------------------------------------

        //-------------------------------------------------------------------------------------------
        try {
            System.out.println("Before exception");
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(Exception e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
    }
    public void cse1(int x, int y) {
```

```
<terminated> Main (2) [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\ja
Before exception
The divide by zero exception
e = java.lang.ArithmeticException: / by zero
e.printStackTrace() =
java.lang.ArithmeticException: / by zero
	at org.studyeasy.Main.main(Main.java:19)
```

The Divide by zero error occurs - so further executation does not happen. This is called Partial executation

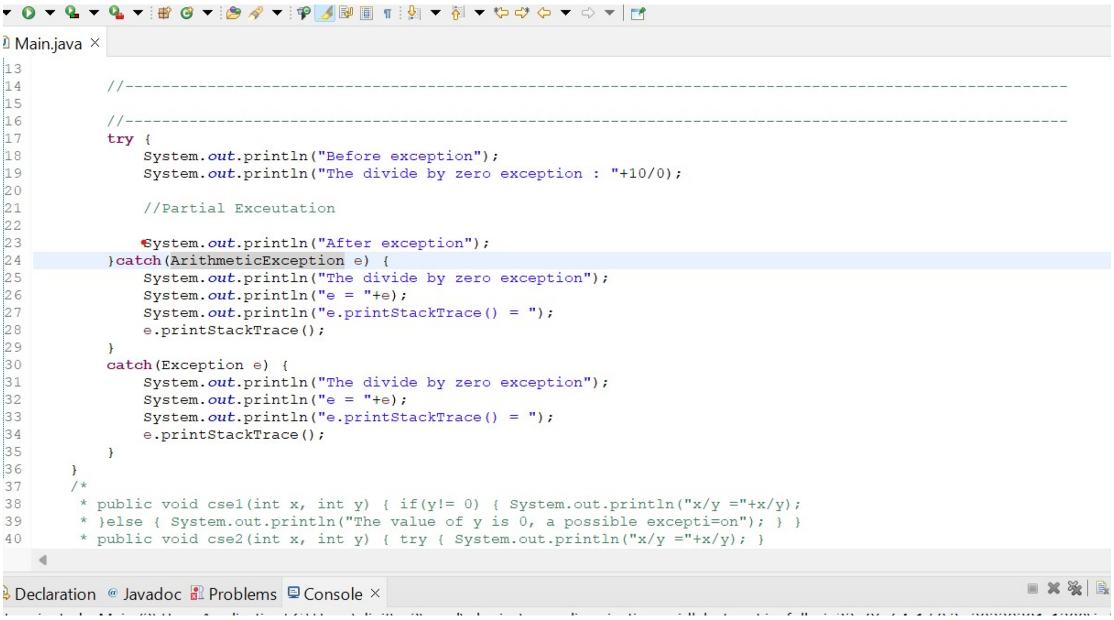
- Can we add multiple catch blocks?

- Every Catch Block is unique.Else you get an error like below.

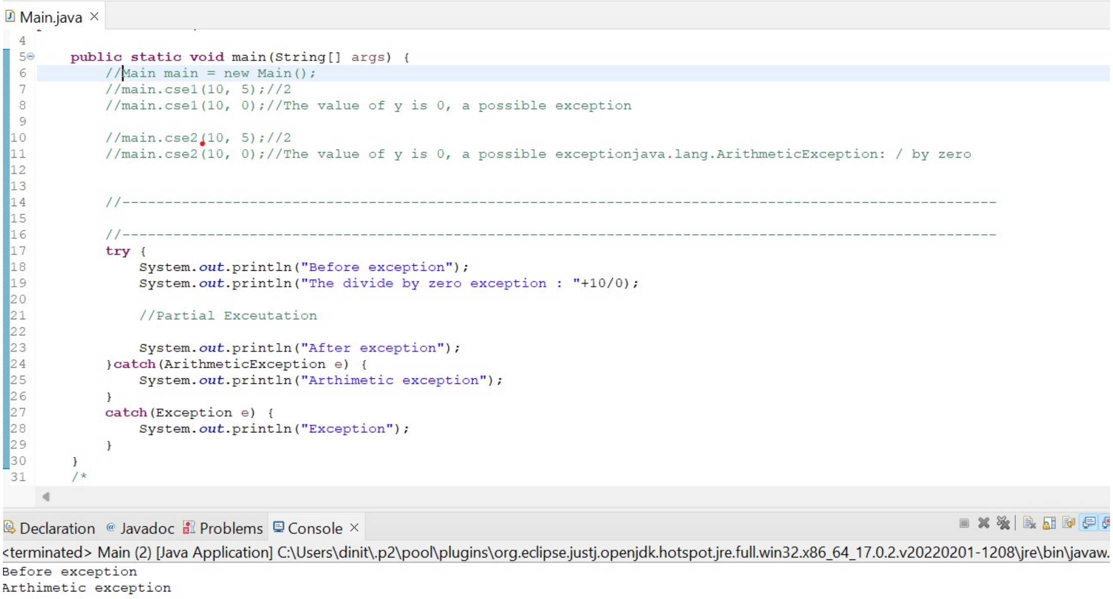


```
        //-----------------------------------------------------------------------

        //-----------------------------------------------------------------------
        try {
            System.out.println("Before exception");
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(Exception e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
        catch(Exception e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
    }
    /*
     * public void cse1(int x, int y) { if(y!= 0) { System.out.println("x/y ="+x
     * }else { System.out.println("The value of y is 0, a possible excepti=on");
     * public void cse2(int x, int y) { try { System.out.println("x/y ="+x/y); }
```

- Solution is below.



```
        //-----------------------------------------------------------------------

        //-----------------------------------------------------------------------
        try {
            System.out.println("Before exception");
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(ArithmeticException e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
        catch(Exception e) {
            System.out.println("The divide by zero exception");
            System.out.println("e = "+e);
            System.out.println("e.printStackTrace() = ");
            e.printStackTrace();
        }
    }
    /*
     * public void cse1(int x, int y) { if(y!= 0) { System.out.println("x/y ="+x/y);
     * }else { System.out.println("The value of y is 0, a possible excepti=on"); } }
     * public void cse2(int x, int y) { try { System.out.println("x/y ="+x/y); }
```

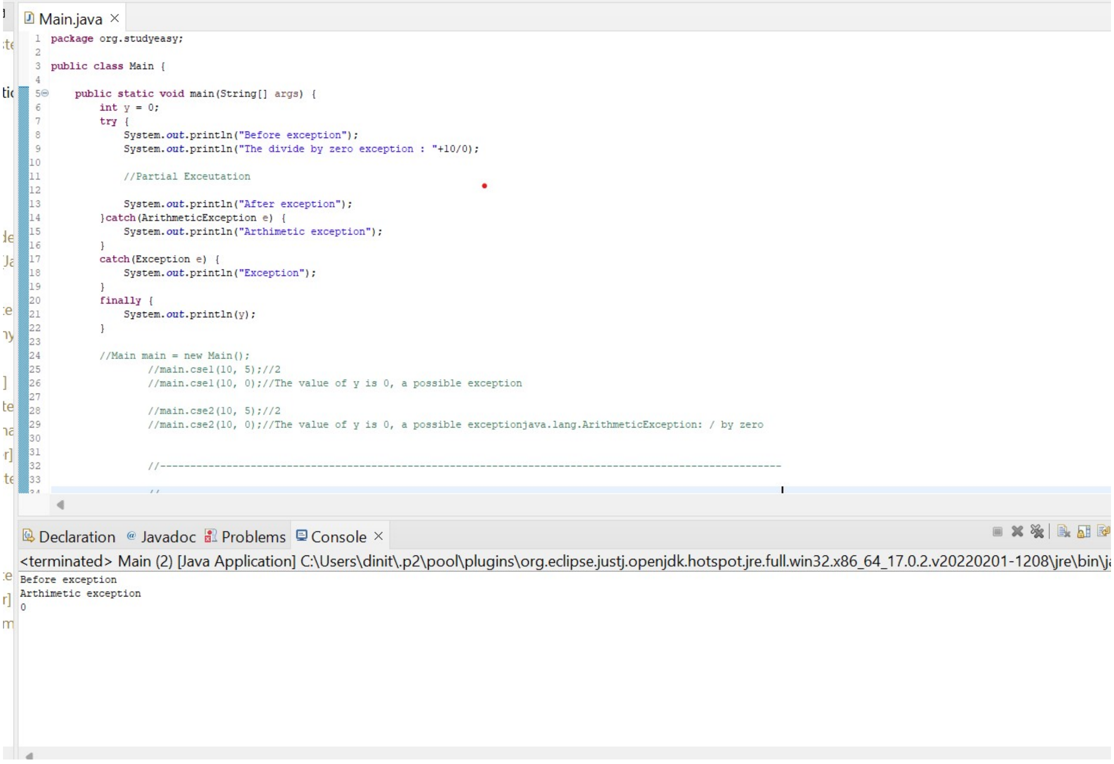


```
    public static void main(String[] args) {
        //Main main = new Main();
        //main.cse1(10, 5);//2
        //main.cse1(10, 0);//The value of y is 0, a possible exception

        //main.cse2(10, 5);//2
        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero


        //----------------------------------------------------------------------------------------------------------

        //----------------------------------------------------------------------------------------------------------
        try {
            System.out.println("Before exception");
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(ArithmeticException e) {
            System.out.println("Arthimetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
    }
    /*
```

```
<terminated> Main (2) [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.
Before exception
Arthimetic exception
```

- ➢ Exception can be caught by one exception. Sequence of Exception does matter a lot.We can have any no of catch blocks.

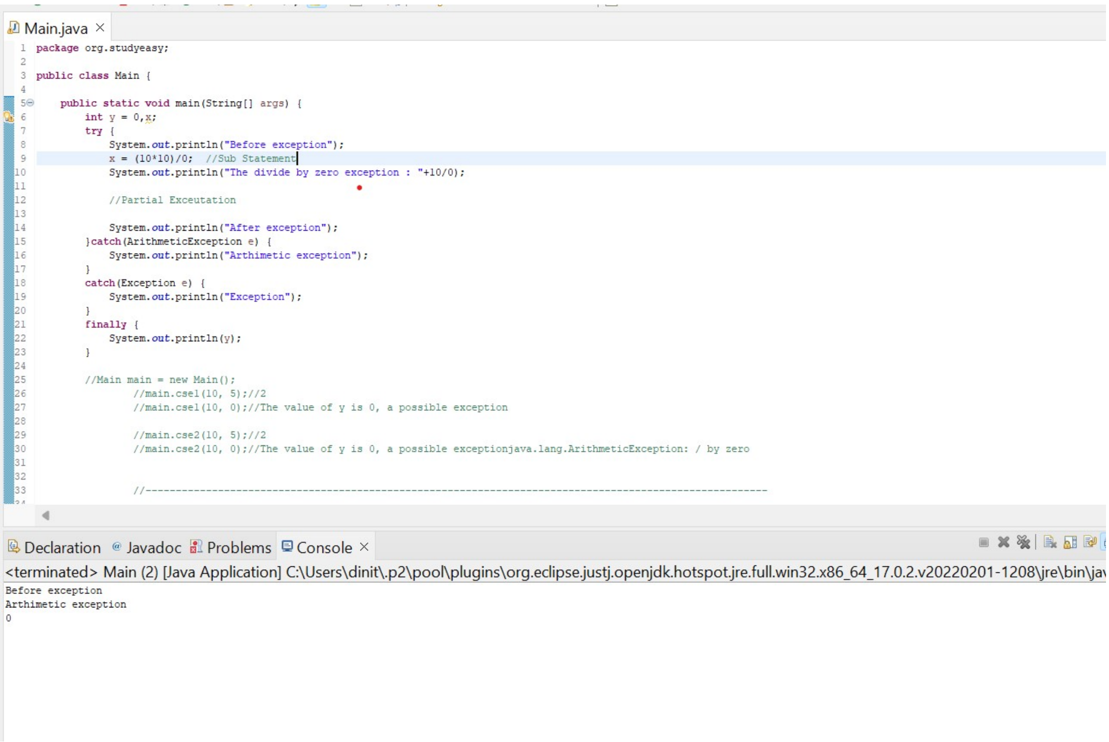
- ➢ **Finally** keyword - It does not matter whether there is exception or not, finally always gets executed.Below is the example.



```
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0;
        try {
            System.out.println("Before exception");
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(ArithmeticException e) {
            System.out.println("Arthimetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
        finally {
            System.out.println(y);
        }

        //Main main = new Main();
                //main.cse1(10, 5);//2
                //main.cse1(10, 0);//The value of y is 0, a possible exception

                //main.cse2(10, 5);//2
                //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero


                //----------------------------------------------------------------------------------------------------
```

```
<terminated> Main (2) [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\j
Before exception
Arthimetic exception
0
```

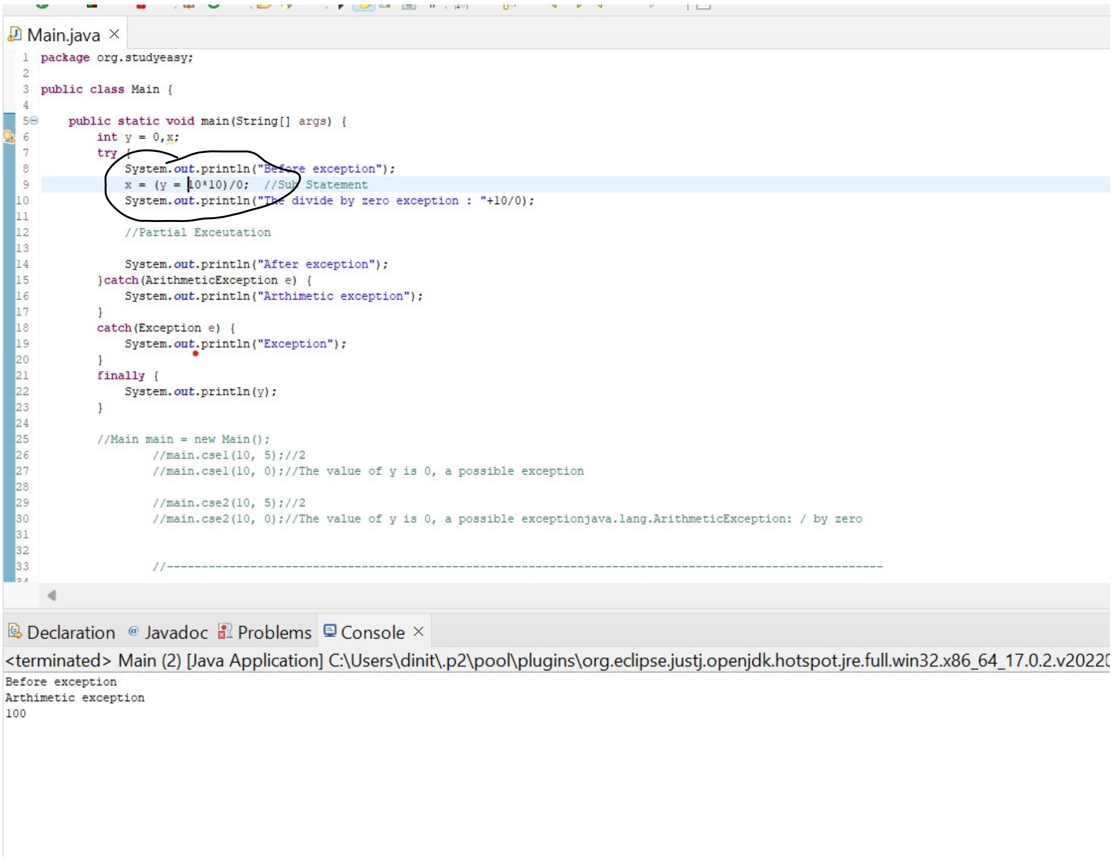
- ➢ Sub Statement



```java
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (10*10)/0;  //Sub Statement
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(ArithmeticException e) {
            System.out.println("Arthimetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
        finally {
            System.out.println(y);
        }

        //Main main = new Main();
                //main.csel(10, 5);//2
                //main.csel(10, 0);//The value of y is 0, a possible exception

                //main.cse2(10, 5);//2
                //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero


                //-----------------------------------------------------------------------------------------------------
```

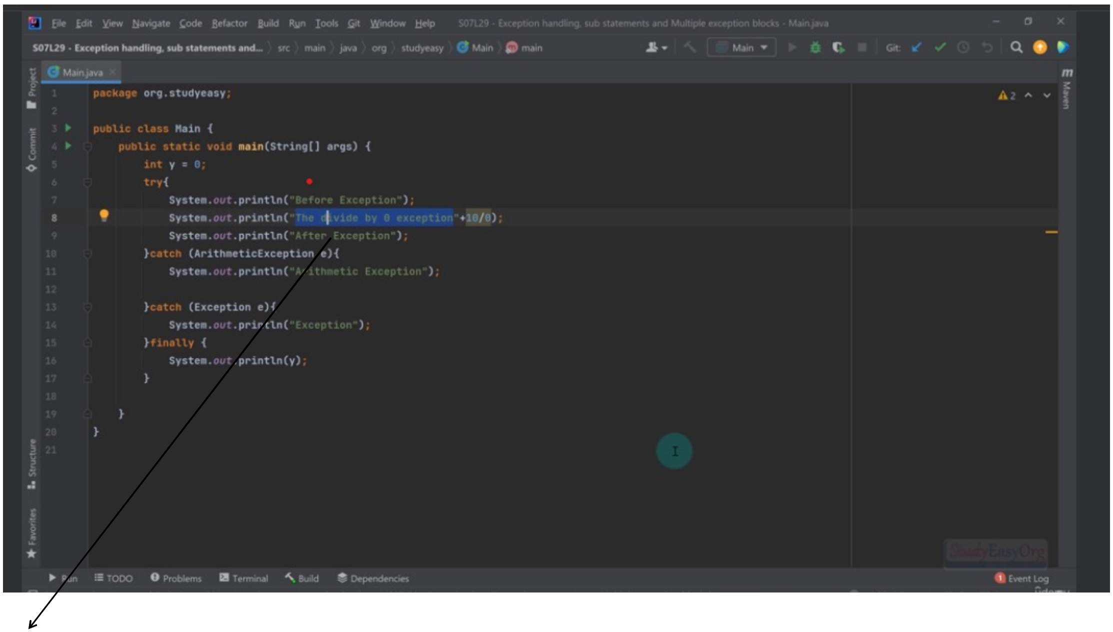
```
<terminated> Main (2) [Java Application] C:\Users\dinit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\jav
Before exception
Arthimetic exception
0
```

- If there are sub statements before an exception, those will get actually executed.May end up giving different result. Sub Statement will get executed even if it is in the same line where we have the exception.



```java
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/0;  //Sub Statement
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(ArithmeticException e) {
            System.out.println("Arthimetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
        finally {
            System.out.println(y);
        }

        //Main main = new Main();
                //main.csel(10, 5);//2
                //main.csel(10, 0);//The value of y is 0, a possible exception

                //main.cse2(10, 5);//2
                //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero


                //-----------------------------------------------------------------------------------------------------
```

```
<terminated> Main (2) [Java Application] C:\Users\dinit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v2022C
Before exception
Arthimetic exception
100
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
S07L29 - Exception handling, sub statements and Multiple exception blocks - Main.java
S07L29 - Exception handling, sub statements and... src main java org studyeasy Main main
Main.java
Run TODO Problems Terminal Build Dependencies
Event Log

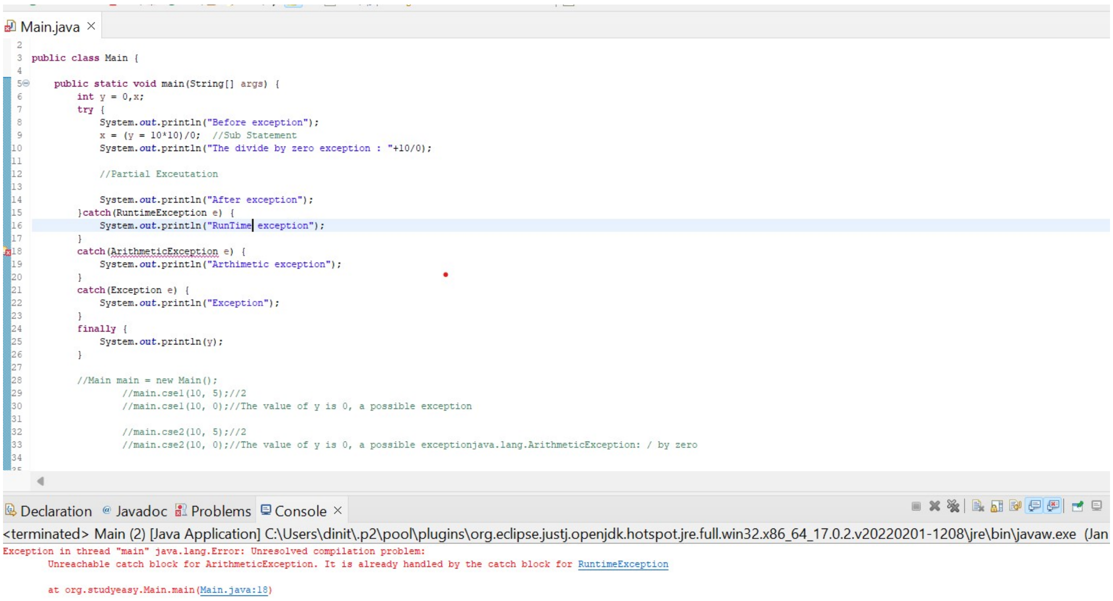
```java
package org.studyeasy;

public class Main {
    public static void main(String[] args) {
        int y = 0;
        try{
            System.out.println("Before Exception");
            System.out.println("The divide by 0 exception"+10/0);
            System.out.println("After Exception");
        }catch (ArithmeticException e){
            System.out.println("Arithmetic Exception");

        }catch (Exception e){
            System.out.println("Exception");
        }finally {
            System.out.println(y);
        }

    }
}
```

Not a Sub Statement - Part of a complete statement.

- ➢ <u>Sequencing of Catch Blocks and Finally Block</u>

- ✓ Child exception should be always before the Parent Exception and if sequencing is not correct, you will get an issue.

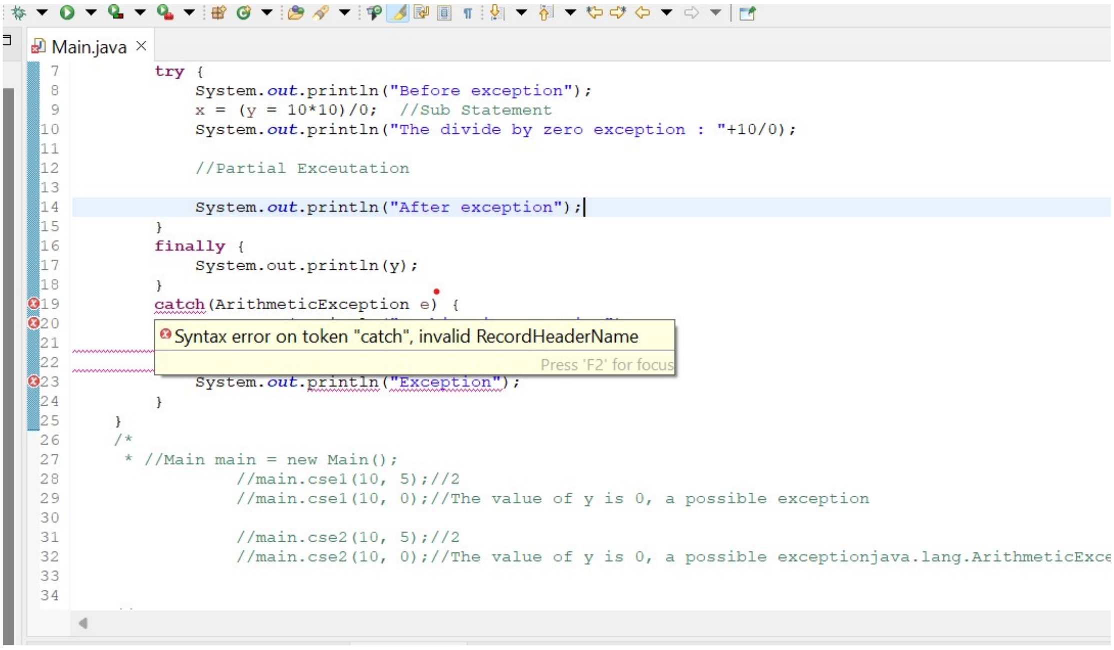
- ✓ Example: RunTime Exception is parent of Arthimetic Exception.



```java

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/0;  //Sub Statement
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }catch(RuntimeException e) {
            System.out.println("RunTime exception");
        }
        catch(ArithmeticException e) {
            System.out.println("Arthimetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
        finally {
            System.out.println(y);
        }

        //Main main = new Main();
            //main.cse1(10, 5);//2
            //main.cse1(10, 0);//The value of y is 0, a possible exception

            //main.cse2(10, 5);//2
            //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero

```

```
<terminated> Main (2) [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
Exception in thread "main" java.lang.Error: Unresolved compilation problem: 
	Unreachable catch block for ArithmeticException. It is already handled by the catch block for RuntimeException

	at org.studyeasy.Main.main(Main.java:18)
```

- ✓ Finally block always have to be the last one.Else it will give a runtime error.



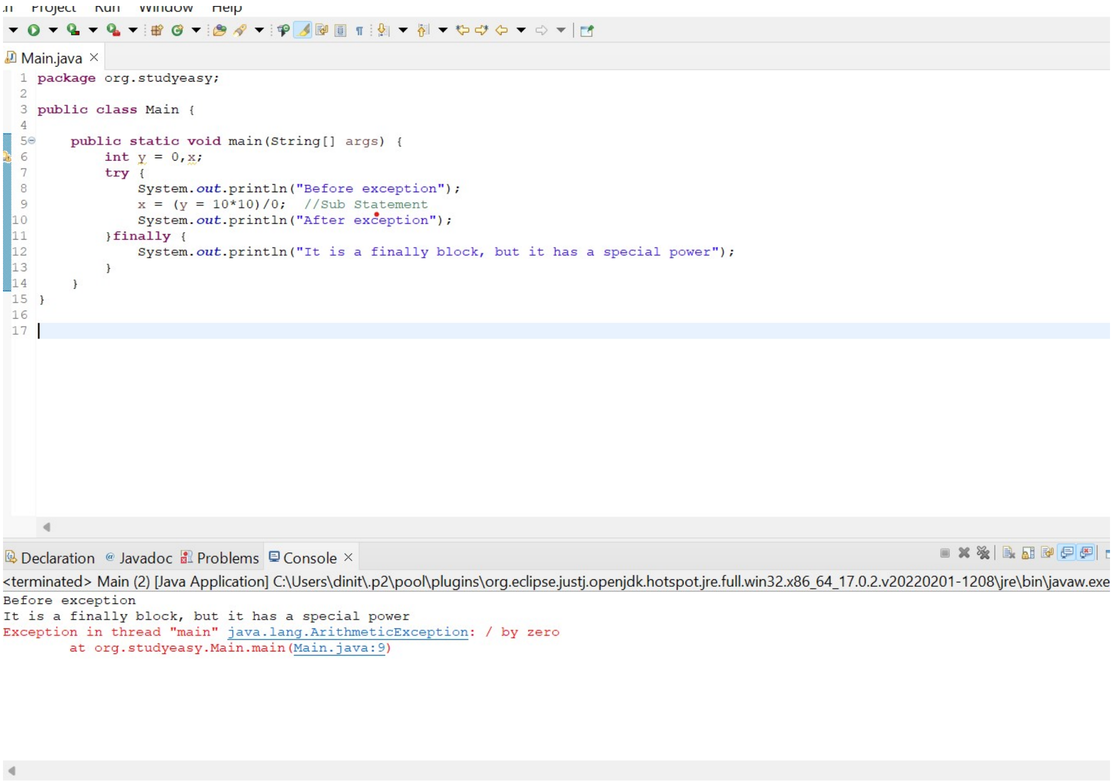
```
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/0;   //Sub Statement
            System.out.println("The divide by zero exception : "+10/0);

            //Partial Exceutation

            System.out.println("After exception");
        }
        finally {
            System.out.println(y);
        }
        catch(ArithmeticException e) {
```

Syntax error on token "catch", invalid RecordHeaderName

Press 'F2' for focus

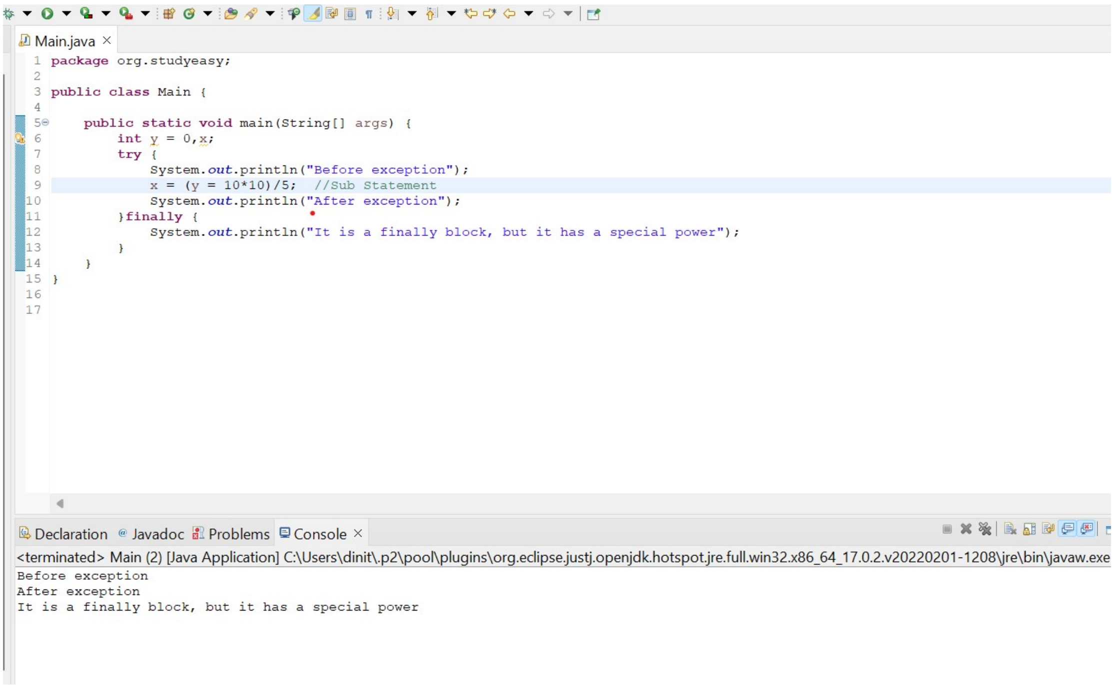
```
            System.out.println("Exception");
        }
    }
    /*
     * //Main main = new Main();
             //main.cse1(10, 5);//2
             //main.cse1(10, 0);//The value of y is 0, a possible exception

             //main.cse2(10, 5);//2
             //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticExce
```

- ## Special Power of finally block



```
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/0;  //Sub Statement
            System.out.println("After exception");
        }finally {
            System.out.println("It is a finally block, but it has a special power");
        }
    }
}
```

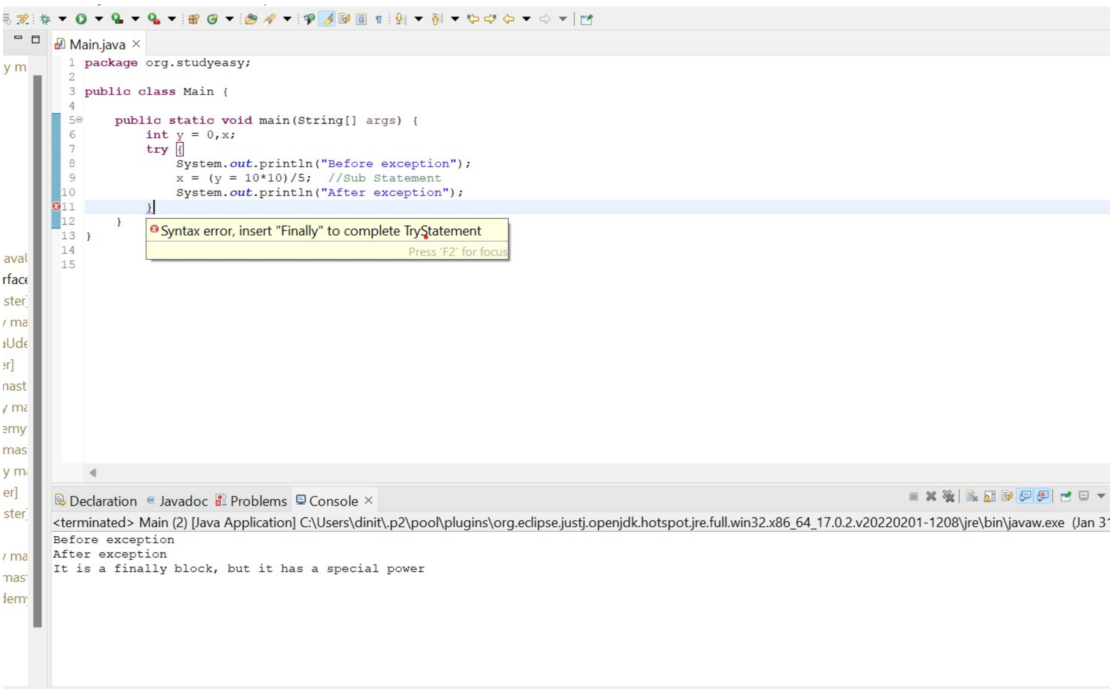
```
<terminated> Main (2) [Java Application] C:\Users\dinit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\javaw.exe
Before exception
It is a finally block, but it has a special power
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at org.studyeasy.Main.main(Main.java:9)
```



```java
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/5;  //Sub Statement
            System.out.println("After exception");
        }finally {
            System.out.println("It is a finally block, but it has a special power");
        }
    }
}
```

```
<terminated> Main (2) [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
Before exception
After exception
It is a finally block, but it has a special power
```

- <u>Without Catch Block or finally - you get an error</u>



```java
package org.studyeasy;

public class Main {

    public static void main(String[] args) {
        int y = 0,x;
        try {
            System.out.println("Before exception");
            x = (y = 10*10)/5;  //Sub Statement
            System.out.println("After exception");
        }
    }
}
```

Syntax error, insert "Finally" to complete TryStatement

Press 'F2' for focus

```
<terminated> Main (2) [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 3
Before exception
After exception
It is a finally block, but it has a special power
```