

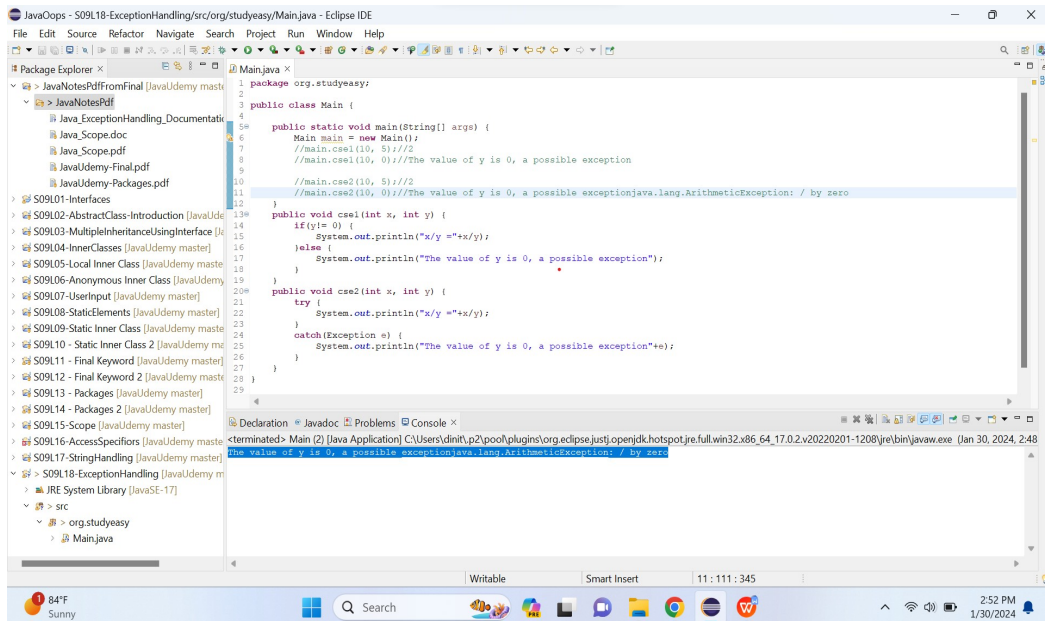
● EXCEPTION HANDLING

- **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



```
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);
        }
    }
}
```

Declaration Javadoc Problems Console x

terminated> Main (2) [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 30, 2024, 2:48 PM)

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.

```
Use Step Filters (Shift+F5)

1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         Main main = new Main();
7         //main.cse1(10, 5);//2
8         //main.cse1(10, 0);//The value of y is 0, a possible exception
9
10        //main.cse2(10, 5);//2
11        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by
12
13        //-----
14        //-----
15
16        try {
17            System.out.println("The divide by zero exception : "+10/0);
18        } catch (Exception e) {
19            System.out.println("The divide by zero exception");
20            System.out.println("e = "+e);
21            System.out.println("e.printStackTrace() = ");
22            e.printStackTrace();
23        }
24    }
25
26    public void cse1(int x, int y) {
27        if (y != 0) {
28            System.out.println("x/y =" + x/y);
29        }
30    }
31}

Declaration @ Javadoc Problems Console x
<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?

```
Main.java x
4
5     public static void main(String[] args) {
6         Main main = new Main();
7         //main.cse1(10, 5);//2
8         //main.cse1(10, 0);//The value of y is 0, a possible exception
9
10        //main.cse2(10, 5);//2
11        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero
12
13        //-----
14        //-----
15
16        try {
17            System.out.println("Before exception");
18            System.out.println("The divide by zero exception : "+10/0);
19
20            //Partial Execution
21
22            System.out.println("After exception");
23        } catch (Exception e) {
24            System.out.println("The divide by zero exception");
25            System.out.println("e = "+e);
26            System.out.println("e.printStackTrace() = ");
27            e.printStackTrace();
28        }
29    }
30
31    public void cse1(int x, int y) {
32        if (y != 0) {
33            System.out.println("x/y =" + x/y);
34        }
35    }
36}

Declaration @ Javadoc Problems Console x
<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
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 execution does not happen. This is called Partial execution

➤ Can we add multiple catch blocks?

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

Main.java ×

```
3
1 //-----
5 //-----
7 try {
3   System.out.println("Before exception");
9   System.out.println("The divide by zero exception : "+10/0);
1
   //Partial Exceutation
2
3   System.out.println("After exception");
1   } catch (Exception e) {
5     System.out.println("The divide by zero exception");|
5     System.out.println("e = "+e);
7     System.out.println("e.printStackTrace() = ");
3     e.printStackTrace();
9   }
1   catch (Exception e) {
1     System.out.println("The divide by zero exception");
2     System.out.println("e = "+e);
3     System.out.println("e.printStackTrace() = ");
1     e.printStackTrace();
5   }
5 }
7 /*
3 * public void cse1(int x, int y) { if(y!= 0) { System.out.println("x/y =" +x
9 * }else { System.out.println("The value of y is 0, a possible excepti=on");
1 * public void cse2(int x, int y) { try { System.out.println("x/y =" +x/y); }
```

- Solution is below.

Main.java ×

```
13
14 //-----
15 //-----
16
17 try {
18   System.out.println("Before exception");
19   System.out.println("The divide by zero exception : "+10/0);
20
21   //Partial Exceutation
22
23   System.out.println("After exception");
24 } catch (ArithmeticException e) {
25   System.out.println("The divide by zero exception");
26   System.out.println("e = "+e);
27   System.out.println("e.printStackTrace() = ");
28   e.printStackTrace();
29 }
30 catch (Exception e) {
31   System.out.println("The divide by zero exception");
32   System.out.println("e = "+e);
33   System.out.println("e.printStackTrace() = ");
34   e.printStackTrace();
35 }
36 }
37 /*
38 * public void cse1(int x, int y) { if(y!= 0) { System.out.println("x/y =" +x/y);
39 * }else { System.out.println("The value of y is 0, a possible excepti=on"); } }
40 * public void cse2(int x, int y) { try { System.out.println("x/y =" +x/y); }
```

Declaration Javadoc Problems Console ×

```

Main.java ×
4
5 public static void main(String[] args) {
6     //Main main = new Main();
7     //main.cse1(10, 5); //2
8     //main.cse1(10, 0); //The value of y is 0, a possible exception
9
10    //main.cse2(10, 5); //2
11    //main.cse2(10, 0); //The value of y is 0, a possible exception java.lang.ArithmeticException: / by zero
12
13    //-----
14    //-----
15
16    try {
17        System.out.println("Before exception");
18        System.out.println("The divide by zero exception : "+10/0);
19
20        //Partial Execution
21
22        System.out.println("After exception");
23    } catch (ArithmeticException e) {
24        System.out.println("Arithmetic exception");
25    }
26    catch (Exception e) {
27        System.out.println("Exception");
28    }
29
30 }
31 /*

```

Declaration Javadoc Problems Console ×

```

<terminated> Main (2) [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.
Before exception
Arithmetic 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.

```

Main.java ×
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0;
7         try {
8             System.out.println("Before exception");
9             System.out.println("The divide by zero exception : "+10/0);
10
11            //Partial Execution
12
13            System.out.println("After exception");
14        } catch (ArithmeticException e) {
15            System.out.println("Arithmetic exception");
16        }
17        catch (Exception e) {
18            System.out.println("Exception");
19        }
20        finally {
21            System.out.println(y);
22        }
23
24        //Main main = new Main();
25        //main.cse1(10, 5); //2
26        //main.cse1(10, 0); //The value of y is 0, a possible exception
27
28        //main.cse2(10, 5); //2
29        //main.cse2(10, 0); //The value of y is 0, a possible exception java.lang.ArithmeticException: / by zero
30
31        //-----
32        //-----
33    }
34 }

```

Declaration Javadoc Problems Console ×

```

<terminated> Main (2) [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.
Before exception
Arithmetic exception
0

```

- **Sub Statement**

```
Main.java x
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0,x;
7         try {
8             System.out.println("Before exception");
9             x = (10*10)/0; //Sub Statement
10            System.out.println("The divide by zero exception : "+10/0);
11
12            //Partial Execution
13
14            System.out.println("After exception");
15        }catch(ArithmeticException e) {
16            System.out.println("Arithmetic exception");
17        }
18        catch(Exception e) {
19            System.out.println("Exception");
20        }
21        finally {
22            System.out.println(y);
23        }
24
25        //Main main = new Main();
26        //main.cse1(10, 5);//2
27        //main.cse1(10, 0);//The value of y is 0, a possible exception
28
29        //main.cse2(10, 5);//2
30        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero
31
32        //-----
33    }
34}

Declaration @ Javadoc Problems Console x
<terminated> Main (2) [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
Before exception
Arithmetic 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.

```
Main.java x
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0,x;
7         try {
8             System.out.println("Before exception");
9             x = (y = 10*10)/0; //Sub Statement
10            System.out.println("The divide by zero exception : "+10/0);
11
12            //Partial Execution
13
14            System.out.println("After exception");
15        }catch(ArithmeticException e) {
16            System.out.println("Arithmetic exception");
17        }
18        catch(Exception e) {
19            System.out.println("Exception");
20        }
21        finally {
22            System.out.println(y);
23        }
24
25        //Main main = new Main();
26        //main.cse1(10, 5);//2
27        //main.cse1(10, 0);//The value of y is 0, a possible exception
28
29        //main.cse2(10, 5);//2
30        //main.cse2(10, 0);//The value of y is 0, a possible exceptionjava.lang.ArithmeticException: / by zero
31
32        //-----
33    }
34}

Declaration @ Javadoc Problems Console x
<terminated> Main (2) [Java Application] C:\Users\dinit\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v2022
Before exception
Arithmetic exception
100
```



```

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.

➤ Sequencing of Catch Blocks and Finally Block

- ✓ 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 Arithmetic Exception.

```

public class Main {
    public static void main(String[] args) {
        int y = 0;
        try {
            System.out.println("Before exception");
            x = (y * 10) / 0; //Sub Statement
            System.out.println("The divide by zero exception : "+10/0);
            //Partial Execution
        }
        System.out.println("After exception");
        catch(RuntimeException e) {
            System.out.println("Runtime exception");
        }
        catch(ArithmeticException e) {
            System.out.println("Arithmetic exception");
        }
        catch(Exception e) {
            System.out.println("Exception");
        }
        finally {
            System.out.println(y);
        }
    }
}

```

Declaration @ Javadoc Problems Console

<terminated> Main (2) [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

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.

```
7      try {
8          System.out.println("Before exception");
9          x = (y = 10*10)/0; //Sub Statement
10         System.out.println("The divide by zero exception : "+10/0);
11
12         //Partial Execution
13
14         System.out.println("After exception");
15     }
16     finally {
17         System.out.println(y);
18     }
19     catch(ArithmeticException e) {
20         System.out.println("Exception");
21     }
22 }
23
24
25
26
27
28
29
30
31
32
33
34
```

Syntax error on token "catch", invalid RecordHeaderName

➤ Special Power of finally block

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0,x;
7         try {
8             System.out.println("Before exception");
9             x = (y = 10*10)/0; //Sub Statement
10            System.out.println("After exception");
11        }finally {
12            System.out.println("It is a finally block, but it has a special power");
13        }
14    }
15 }
16
17
```

Declaration Javadoc Problems Console

<terminated> Main (2) [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

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)

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0,x;
7         try {
8             System.out.println("Before exception");
9             x = (y = 10*10)/5; //Sub Statement
10            System.out.println("After exception");
11        }finally {
12            System.out.println("It is a finally block, but it has a special power");
13        }
14    }
15 }
16
17
```

Declaration Javadoc Problems Console

<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

➤ Without Catch Block or finally - you get an error

```
1 package org.studyeasy;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         int y = 0,x;
7         try {
8             System.out.println("Before exception");
9             x = (y = 10*10)/5; //Sub Statement
10            System.out.println("After exception");
11        }
12    }
13 }
14
15
```

Syntax error, insert "Finally" to complete TryStatement
Press 'F2' for focus

Declaration Javadoc Problems Console

<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