

Exception handling in Java

Rules:

Can we use try block alone without Catch or finally? the answer is no

Try block alone should not work when there is try block there must be a catch or finally should be present

When there is a catch it should contain one keyword as Exception with an identifier as e catch (Exception e) i.e., there is nothing the identifier should be only as e we can give any character or string pfb., rules and definitions of identifier

What is an identifier in java?

Identifiers are the names of variables, methods, classes, packages and interfaces. Unlike literals they are not the things themselves, just ways of referring to them. In the HelloWorld program, HelloWorld , String , args , main and println are **identifiers**.

Example 1: try block with finally

```
1 public class TryCatch{
2
3     public static void main (String args[])
4     {
5         try{
6
7             int [] numbers={10,20,30};
8
9             System.out.println(numbers[10]);
10        }
11
12        finally {
13
14            System.out.print("i am finally block");
15        }
16    }
17 }
```

Result

```
$javac TryCatch.java
$java -Xmx128M -Xms16M TryCatch
i am finally blockException in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10
    at TryCatch.main(TryCatch.java:9)
```

Example 2: try with catch block

```
1 ~ public class TryCatch{
2
3     public static void main (String args[])
4     {
5         try{
6
7             int [] numbers={10,20,30};
8
9             System.out.println(numbers[10]);
10        }
11
12 ~    catch(Exception e){
13
14        System.out.print("something went wrong");
15    }
16 }
17 }
```

Result

```
$javac TryCatch.java
$java -Xmx128M -Xms16M TryCatch
something went wrong
```

Example 3: Try catch with finally block

```
1 public class TryCatch{
2
3     public static void main (String args[])
4     {
5         try{
6
7             int [] numbers={10,20,30};
8
9             System.out.println(numbers[10]);
10        }
11
12        catch(Exception e){
13
14            System.out.println("something went wrong");
15        }
16
17        finally{
18
19            System.out.println("I am the finally block i will always execute");
20        }
21    }
22 }
```

Result:

```
$javac TryCatch.java
$java -Xmx128M -Xms16M TryCatch
something went wrong
I am the finally block i will always execute
```

Why throw keyword is present is exception handling concept, to display custom error we are using throw keyword

Example 4: throw keyword in Exception handling

When the person is beyond 75 kgs he is eligible for military selection

Created the below snippet to get an intentional error: by seeing description throw is used for custom error I gave my own exception name as AgelimitException but it will not work why because in java libraries they mentioned few general exception type by create our own custom exception we should use any one of these

ArithmeticException, FileNotFoundException, ArrayIndexOutOfBoundsException, SecurityException

```
1 public class MilitarySel{
2
3     public int personweight(int weight)
4     {
5
6         if (weight>75){
7
8             throw new AgelimitException("Person age should be less than or eq1 to
              75 kgs to get selected into military");
9         }
10
11        else
12        {
13            System.out.println("You are not fit for military selection this year
              try next year ");
14        }
15
16    }
17
18    public static void main (String args[])
19    {
20        MilitarySel ms = new MilitarySel();
21
22        ms.personweight(76);
23
24    }
25 }
```

Result:

```
$javac MilitarySel.java
MilitarySel.java:8: error: cannot find symbol
        throw new AgelimitException("Person age should be less than or eq1 to 75 kgs to g
                   ^
    symbol:   class AgelimitException
    location: class MilitarySel
1 error
```

I am going to replace the exception name as Arithmetic Exception by removing AgeLimitException in the above program

Happy path scenario

```
1 public class MilitarySel{
2
3     public int personweight(int weight)
4     {
5
6         if (weight>75){
7
8             throw new ArithmeticException("Person age should be less than or eql
9                 to 75 kgs to get selected into military");
10        }
11        else
12        {
13            System.out.println("You are not fit for military selection this year
14                try next year ");
15        }
16        return 0;
17    }
18
19    public static void main (String args[])
20    {
21        MilitarySel ms = new MilitarySel();
22
23        ms.personweight(71);
24
25    }
26 }
```

Result:

```
Result
$javac MilitarySel.java
$java -Xmx128M -Xms16M MilitarySel
You are not fit for military selection this year try next year
```

Negative Scenario

```
1 public class MilitarySel{
2
3     public int personweight(int weight)
4     {
5
6         if (weight>75){
7
8             throw new ArithmeticException("Person age should be less than or eq1
9                 to 75 kgs to get selected into military");
10
11         }
12     else
13     {
14         System.out.println("You are not fit for military selection this year
15             try next year ");
16
17     }
18
19     return 0;
20 }
21
22 public static void main (String args[])
23 {
24     MilitarySel ms = new MilitarySel();
25
26     ms.personweight(89);
27 }
28 }
```

Result

```
$javac MilitarySel.java
```

```
$java -Xmx128M -Xms16M MilitarySel
```

```
Exception in thread "main" java.lang.ArithmeticException: Person age should be less than c
    at MilitarySel.personweight(MilitarySel.java:8)
    at MilitarySel.main(MilitarySel.java:23)
```

MilitarySel

```
Exception in thread "main" java.lang.ArithmeticException: Person age
should be less than or eq1 to 75 kgs to get selected into military
    at MilitarySel.personweight(MilitarySel.java:8)
    at MilitarySel.main(MilitarySel.java:23)
```