Lecture 12

Can we handle more then one Exception in single ~~else~~ catch block

1)
```
import java.io.*;
class Demo{
public static void main(String ar[])
{
 int x = Integer.parseInt(ar[0]);
 int y = Integer.parseInt(ar[1]);
  System.out.println(x/y);
  System.out.println("Softwaves_2..");
}}
```

o/p → case Demo 10 2	10 0	10 abhi
→ 5	→ Ex: ArithmeticException:	→ Ex: NFE: For
Softwaves_2	\ by zero	input String "abhi"

2)
```
import java.io.*;
class Demo{
public static void main(String ar[])
{
 try{
    int x = Integer.parseInt(ar[0]);
    int y = Integer.parseInt(ar[1]);
   System.out.println(x/y);
 }
 catch(ArithmeticException e)
 {
   System.out.println(e);
 }
  System.out.println("Softwaves_2");
 }
}
```

o/p → 10 2	10 0	10 abhi
5	AE: / by zero	Excep: NFE: for input
Softwaves2	Softwaves_2	String "abhi"

| -> Pipe Symbol

3) import java.io.*;
class Demo{
public static void main(String ar[])
{
try{
int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException e)
{ S.o.p(e);
}
catch(NumberFormatException e)
{
System.out.println(e);
}
System.out.print("Softwaves_2");
}
}

o/p -> 10 2
5
Softwaves_2

10 0
AE: / by zero
Softwaves_2

10 abhi
NFE: for I/P String "abhi"
Softwaves_2

4) import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException | NumberFormateException e){
S.o.p(e);
}
S.o.p("softwaves_222");
}}

o/p -> 10 2
5
softwaves_222

10 0
AE: \ by zero
softwaves_222

10 abhi
NFE: for I/P string "abhi"
softwaves_222

Note

1> we can handle multiple exception with one catch block using a pipe(|).

5]
```
import java.io.*;
class Demo{
public static void main(String ar[])
{
try{ int x = Integer.parseInt(ar[0]);
     int y = Integer.parseInt(ar[1]);
     System.out.println(x/y);
   }
catch(ArithmeticException | NumberFormateException e){
     System.out.println(e);
   }
     System.out.println("Softwaves_222");
}
}
```

o/p→ 10 2	10 0	10 abhi	10
5	AE: / by zero	NFE: for i/p str "abhi"	Exception: ArrayIndexOutofBounds Exception: 1
Softwaves_222	Softwaves_222	Softwaves_222	

→ Exception class is the child classes of all Exception को handle करने के लिए है।

6]
```
import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
     int y = Integer.parseInt(ar[1]);
     System.out.println(x/y);
   }
catch(Exception e)
{  S.o.p(e);
}
  System.out.println("Softwaves_222");
 }
}
```

o/p→ 10 5	10 0	10 abhi	10
2	AE: \ by zero	NFE: for I/P String	AIOOBE: 1
Softwaves_222	Softwaves-222	"abhi"	Softwaves-222
		Softwaves_222	

1.) import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException e){
System.out.println("abhi");
}
catch(Exception e){ System.out.println("jain"); }
System.out.println("Softwaves_2");
}
}

o/p→ 10 2	10 0	10 abhi	10
5	abhi	jain	jain
Softwaves_2	Softwaves_2	Softwaves_2	Softwaves_2

8) import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(Exception e){ System.out.println("jain"); }
catch(ArithmeticException e){ S.o.p.ln("abhi"); }
System.out.println("Softwaves_2");
}
}

o/p→ Error: exception ArithmeticException has already been caught

o/p → 10 5	10 0	10 abhi	10
2	AE: \ by zero	NFE: for I/P string	AIOOBE: 1
Softwaves_222	Softwaves-222	"abhi"	Softwaves-222
		Softwaves_222	

1,7] import java.io.*;
Class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException e){
System.out.println("abhi");
}
catch(Exception e){ System.out.println("jain"); }
System.out.println("Softwaves-2");
}
}

→ 10 2	10 0	10 abhi	10
5	abhi	jain	jain
softwaves-2	softwaves_2	softwaves_2	softwaves_2

8] import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(Exception e){ System.out.println("jain"); }
catch(ArithmeticException e){ S.o.pln("abhi"); }
System.out.println("Softwaves-2");
}
}

o/p → Error: exception ArithmeticException has already been caught

O/P→ 10 5	10 0	10 abhi	10
2	AE: \ by zero	NFE: for I/P String	AIOOBE: 1
Softwaves_222	Softwaves_222	"abhi"	softwaves_222
		softwaves_222	

1,7] import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException e){
System.out.println("abhi");
}
catch(Exception e){ System.out.println("jain"); }
System.out.println("Softwaves_2");
}
}

O/P→ 10 2	10 0	10 abhi	10
5	abhi	jain	jain
softwaves_2	softwaves_2	softwaves_2	softwaves_2

8] import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(Exception e){ System.out.println("jain"); }
catch(ArithmeticException e){ S.o.p.ln("abhi"); }
System.out.println("softwaves_2");
}
}

O/P→ error: exception ArithmeticException has already been caught

2) import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException e){
System.out.println("jain");
}
catch(ArithmeticException e){
System.out.println("abhi");
}
System.out.println("softwaves_2");
}
}

o/p → error: exception ArithmeticException has already been caught

Note

2) when we use multiple catch block with one try block then we can use super class only in last catch block. because if use it any other place then it will give error because super class can handle all the exception.

3) import java.io.*;
class Demo{
public static void main(String ar[]){
try{ int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(ArithmeticException/Exception e){
S.o.pln("jain");
S.o.pln("softwaves_2");
}
}

e = e.toString()

Lacture - 14

o/p → error: Alternatives in a multi-catch statement cannot be related by subclassing.
Alternative ArithmeticException is a subclass of Alternative Exception

Note

3) when we handle multiple exception with one catch block using a pipe(|) then it should not contain any parent-child relationship btween the exceptions.

Lacture 14

Q How many ways to print Exception message

→ 3

1) import java.io.*;
class Demo{
public static void main(String ar[]){
try{
int x = Integer.parseInt(ar[0]);
int y = Integer.parseInt(ar[1]);
System.out.println(x/y);
}
catch(Exception e){ System.out.println(e.toString()); }

System.out.println("Softwaves_2");
}}

o/p → 10 2	10 0	10 abhi
- 5	/by zero	NFE: for I/p st."abhi"
Softwaves_2	Softwaves_2	Softwaves_2

toString() →

→ 10 2	10 0	10 abhi
5	AE: / by zero	NFE: for I/p String "abhi"
Softwaves_2	Softwaves_2	Softwaves_2

diff. ways

1) toString() method : by using this method, we will only get exception name and description of an exception.

2) getMessage() : : by using this method; we will get description of exception.

3) printStackTrace() : by using this method, we will get exception name and description of an exception separated by colon, and stacktrace in the next line

getmessage() =>

2)
```
import java.io.*;
class Demo{
public static void main(String ar[]){
try{
    int x = Integer.parseInt(ar[0]);
    int y = Integer.parseInt(ar[1]);
     System.out.println(x/y);
   }
catch(Exception e){
    System.out.println(e.getMessage());
  }
System.out.println("Softwaves_2");
 }
}
```

o/p → 10 2	10 0	10 abhi
5	\ by zero	for I/P string "abhi"
Softwaves_2	Softwaves_2	Softwaves_2

3) import java.io.*;
class Demo36{
public static void main(String arr[]){
try{
Int x = Integer.parseInt(arr[0]);
Int y = Integer.parseInt(arr[1]);
System.out.println(x/y);
}
catch(Exception e){
~~System.out.println~~ e.printStackTrace();
}
System.out.println("Softwaves_2");
}
}

o/p →	10 2	10 0	10 abhi
	5	AE: / by zero	NFE: for i/p string "abhi"
	Softwaves_2	at Demo36.main();	at Demo36.main()
		Softwaves_2	Softwaves_2

4) class Demo37{
public static void main(String arr[]){
try{
System.out.println(10/0);
}
catch(Exception e){
e.printStackTrace();
}
System.out.println("Softwaves_2");
}

o/p → ~~E~~ E: AE: X by zero
at Demo37.main()

5]
```
class A{
    Void Show1(){
     try{
          System.out.println(10/0);
        }
     catch(Exception c){
        // e.printStackTrace();
          System.out.println(e);
     }}
}

class B{
    Void show2(){
        A a = new A();    a.show1();
    }}

class Demo38{
    public static Void main(String arg[]){
        B b = new B();
          b.show2();
        System.out.println("Softwaves-2");
    }}
```

o/p → E: AE: / by zero
Softwaves-2

6]
```
class A{
    void show1(){
    try{
          System.out.println(10/0);
        }
    catch(Exception e){
          System.out.println(e.toString());
        }
}}

class B{ void show2(){  A a = new A();
          a.show1();
        }
}
```

```
no{
static void main(String arr[])
b = new B();
 b.show2();
tem.out.println("Softwaves_2");

metic Exception:
 zero

s_2

 {
 static void main(String arr[]){

ss A{
id show1(){
try{
    System.out.println(10/0);
  }
 catch(Exception e){
    System.out.println(e.getMessage());
}}}

lass B{
  void show2(){
    A a = new A();
    a.show1();
  }}

class Demo38{
  public static void main(String arr[]){
      B b = new B();
       b.show2();
      System.out.println("Softwaves_2");
  }}

o/p → / by zero
 Softwaves_2
```

```
class Demo{
  public static void main(String arr[])
{   B b = new B();
        b.show2();
    System.out.println("softwaves_2");
  }}
```

o/p → Arithmetic Exception :
/ by zero
softwaves_2

7) ~~class demo {~~
~~public static void main (String arr[]) {~~

7) class A{
  void show1(){
  try{
      System.out.println(10/0);
    }
  catch(Exception e){
      System.out.println(e.getMassage());
  }}}

class B{
  void show2(){
    A a = new A();
    a.show1();
  }}

class Demo38{
  public static void main(String arr[]){
      B b = new B();
      b.show2();
    System.out.println("softwaves_2");
  }}

o/p → / by zero
softwaves_2

Q) class A{
    void show1(){
        try{
            System.out.println(10/0);
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }}

class B{
    void show2(){
        A a = new A();
        a.show1();
    }}

class Demo38{
    public static void main(String args[]){
        B b = new B();
        b.show2();
    }}

O/P→ java.lang.ArithmeticException : / by zero
        at A.show1()
        at B.show2()
        at Demo38.main()
softwaves_2

⇒ जब JVM Exception को main() method को Default Exception Handler को pass करता है तो DefaultExceptionHandler internally printStackTrace() method को use करता है।
इसलिए यह stack की form में print होता है।
massage

```
}}
class B{
    void show2(){
        A a = new A();
        a.show1();
    }
}

class Demo38{
    public static void main(String arg[]){
        B b = new B();
        b.show2();
    }
}
```

O/P → java.lang.ArithmeticException : / by zero
at A.show1()
at B.show2()
at Demo38.main()

softwaves-2

→ JVM Exception को main() method से Default Exception Handler को pass करता है तो DefaultExceptionHandler internally printStackTrace() method का use करता है।

इसलिए यह stack की form में print होता है।

message

Lecture 15

fully checked Exception

- अगर Exception class किसी sub classes में checkedException हो तो ऐसी class को fully checked Exception कहते है।

Ex → IOException

partially checked Exception

the exception which have both checked and unchecked subclasses is called partially checked exception.

Ex → Throwable
→ Exception