

Java Exception Handling Concepts (Part 2)

For

BCA 3rd Semester (Session 2017-20) Students (J.D. Women's College, Patna)

By

Sumit Kumar Verma

Now I am trying to solve above problem using Exception Handling Process . As we know Java provide us five tools like try,catch,finally,throw,throws .

Here we are using try and catch to maintain the normal flow of the program.

```
class Ex1
{
    public static void main(String args[])
    {
        int a=Integer.parseInt(args[0]);
        int b=Integer.parseInt(args[1]);
        try
        {
            System.out.println(a/b);
        }catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        System.out.println("normal flow");
    }
}
```

o/p:1>if we pass 4 2 as input then it give

2

normal flow

2>if we pass 4 0 as input then it give

ArithmeticException

normal flow

- A try block can contain multiple catch block but must be in specific order i.e from child to parent i.e specific to generalize

Program:-

```
import java.util.*;
class MultiCatch
{
    public static void main(String args[])
    {
        Scanner in =new Scanner(System.in);
        System.out.println("Enter the numbers");
        int a=in.nextInt();
        int b=in.nextInt();
        try
        {
            System.out.println(a/b);
        }

        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e);
        }
        catch(Exception e)
        {
```

```

        System.out.println(e);
    }
}

```

Finally :- **finally** keyword is used to define a block of statements that is executed by the JRE with 100% certainty whether an exception occurs or not . it is mainly used to fulfil the last wish of developer . like closing connection with database .

Program:-

```

import java.util.*;
class Finally1
{
    public static void main(String args[])
    {
        Scanner in =new Scanner(System.in);
        System.out.println("Enter the numbers");
        int a=in.nextInt();
        int b=in.nextInt();
        try{
            System.out.println(a/b);
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        finally
        {
            System.out.println("finally");
        }

        System.out.println("normal flow");
    }
}

```

```
    }  
}
```

o/p:1>if we pass 4 2 as input then it give

2

finally

normal flow

2>if we pass 4 0 as input then it give

ArithmeticException divided by zero

finally

normal flow

```
import java.util.*;
```

```
class MultiCatch
```

```
{
```

```
    public static void main(String args[])
```

```
{
```

```
        Scanner in =new Scanner(System.in);
```

```
        System.out.println("Enter the numbers");
```

```
        int a=in.nextInt();
```

```
        int b=in.nextInt();
```

```
        try
```

```
        {
```

```
            System.out.println(a/b);
```

```
        }
```

```
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e);
        }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
}
```

```
import java.util.*;
class Finally1
{
    public static void main(String args[])
    {
        Scanner in =new Scanner(System.in);
        System.out.println("Enter the numbers");
        int a=in.nextInt();
        int b=in.nextInt();
    }
}
```

```
try{
    System.out.println(a/b);
}
catch(ArithmeticException e)
{
    System.out.println(e);
}
finally
{
    System.out.println("finally");
}

System.out.println("normal flow");
}
}
```