

JBK1011-Exception

An example of Try catch in Java

```
class Example1 {  
    public static void main(String args[]) {  
        int num1, num2;  
        try {  
            num1 = 0;  
            num2 = 62 / num1;  
            System.out.println("Try block message");  
        } catch (ArithmeticException e) {  
            System.out.println("Error: Don't divide a number by zero");  
        }  
        System.out.println("I'm out of try-catch block in Java.");  
    }  
}
```

Example:

```
public class Exception_Ex1 {  
    public static void main(String args[]) {  
        int[] array = {1, 2};  
        try {  
            System.out.println("The 10th value of Array is : " + array[10]);  
        }  
        catch (Exception e) {  
            System.out.println("The Error : " + e);  
        }  
    }  
}
```

A try block can have any number of catch blocks.

```
class Example2 {  
    public static void main(String args[]) {  
        try {  
            int a[] = new int[7];  
            a[4] = 30 / 0;  
            System.out.println("First print statement in try block");  
        }  
        catch (ArithmeticException e) {  
            System.out.println("Warning: ArithmeticException");  
        }  
        catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Warning: ArrayIndexOutOfBoundsException");  
        }  
    }  
}
```

```
}  
catch(Exception e){  
    System.out.println("Warning: Some Other exception");  
}  
System.out.println("Out of try-catch block...");  
}}
```

Nested try catch example - explanation

```
class Nest{  
    public static void main(String args[]){  
        //outer try block  
        try{  
            //Inner try block1  
            try{  
                System.out.println("Inside block1");/  
                int b =45/0;  
                System.out.println(b); }  
            catch(ArithmeticException e1){  
                System.out.println("Exception: e1"); }  
            //Inner try block2  
            try{  
                System.out.println("Inside block2");  
                int b =45/0;  
                System.out.println(b);  
            }  
            catch(ArrayIndexOutOfBoundsException e2){  
                System.out.println("Exception: e2"); }  
            System.out.println("Just other statement"); }  
        catch(ArithmeticException e3){  
            System.out.println("Arithmetic Exception");  
            System.out.println("Inside parent try catch block"); }  
        catch(ArrayIndexOutOfBoundsException e4){  
            System.out.println("ArrayIndexOutOfBoundsException");  
            System.out.println("Inside parent try catch block"); }  
        catch(Exception e5){  
            System.out.println("Exception");  
            System.out.println("Inside parent try catch block"); }  
        System.out.println("Next statement..");  
    } }  
}
```

Example of throws Clause

```
class Demo{
```

```
static void throwMethod() throws NullPointerException{
    System.out.println ("Inside throwMethod");
    throw new NullPointerException ("Demo");
}
public static void main(String args[]){
    try{
        throwMethod();
    } catch (NullPointerException exp){
        System.out.println ("The exception get caught" +exp);
    } }
```

"Throws keyword"

Example:

```
import java.io.*;
public class ThrowExample {
    void mymethod(int num)throws IOException,
    ClassNotFoundException{
        if(num==1)
            throw new IOException("Exception Message1");
        else
            throw new ClassNotFoundException("Exception
            Message2");
    }
}
class Demo{
    public static void main(String args[]){
        try{
            ThrowExample obj=new ThrowExample();
            obj.mymethod(1);
        }catch(Exception ex){
            System.out.println(ex);
        } }
```

How to throw your own exception explicitly using throw keyword

```
class MyOwnException extends Exception {
    public MyOwnException(String msg){
        super(msg);
    }
}
class EmployeeTest {
    static void employeeAge(int age) throws MyOwnException{
        if(age < 0)
```

```
        throw new MyOwnException("Age can't be less than zero");
    else
        System.out.println("Input is valid!!");
}
public static void main(String[] args) {
    try {
        employeeAge(-2);
    }
    catch (MyOwnException e) {
        e.printStackTrace();
    } } }
```

How to throw an already defined exception using throw keyword

```
class Exception2{
    static int sum(int num1, int num2){
        if (num1 == 0)
            throw new ArithmeticException("First parameter is not valid");
        else
            System.out.println("Both parameters are correct!!");
        return num1+num2;
    }
    public static void main(String args[]){
        int res=sum(0,12);
        System.out.println(res);
        System.out.println("Continue Next statements");
    } }
```

Using "throw keyword" we can throw checked, unchecked and user -defined exceptions.

```
public class ThrowExample {
    static void checkEligibility(int stuage, int stuweight){
        if(stuage<12 && stuweight<40) {
            throw new ArithmeticException("Student is not eligible for registration");
        }
        else {
            System.out.println("Entries Valid!!");
        } }
    public static void main(String args[]){
```

```
System.out.println("Welcome to the Registration  
process!!");  
checkEligibility(10, 39);  
System.out.println("Have a nice day..");  
}}
```

Multiple Catch block:

```
public class Exception_Ex2 {  
    public static void main (String args[]) {  
        int array[] = {20,10,30};  
        int num1 = 15, num2 = 0;  
        int sum = 0;  
        try{  
            sum = num1 / num2;  
            System.out.println("The result is : " + sum);  
            for(int i =0; i <10; i++){  
                System.out.println("The value of array are" +array[i]);  
            } }  
        catch (ArrayIndexOutOfBoundsException e){  
            System.out.println("Error 1 : " + e);  
        }  
        catch (ArithmeticException e) {  
            System.out.println ("Error 2 : " + e);  
        } } }
```

Create your Own Exception in Java

```
public class OwnException extends Exception {  
    public OwnException(int msg) {  
        super(String.valueOf(msg));  
    } public OwnException(float msg) {  
        super(String.valueOf(msg));  
    } public OwnException(char msg) {  
        super(String.valueOf(msg));  
    } public OwnException(String msg) {  
        super(msg);  
    } }  
class MainClass {  
    public static void intFn() throws OwnException {  
        System.out.println("Throwing OwnException from intFn()");  
        throw new OwnException(10);  
    }  
}
```

```
public static void floatFn() throws OwnException {
    System.out.println("Throwing OwnException from floatFn()");
    throw new OwnException((float) 111.111); }
public static void charFn() throws OwnException {
    System.out.println("Throwing OwnException from charFn()");
    throw new OwnException('A'); }
public static void StringFn() throws OwnException {
    System.out.println("Throwing OwnException from
StringFn()");
    throw new OwnException("Java World..");
} public static void main(String[] args) {
    try {
        intFn();
    } catch (OwnException e) {
        e.printStackTrace();
    } try {
        floatFn();
    } catch (OwnException e) {
        e.printStackTrace();
    } try {
        charFn();
    } catch (OwnException e) {
        e.printStackTrace();
    } try {
        StringFn();
    } catch (OwnException e) {
        e.printStackTrace();
    } } }
```

Finally Block in Java

```
public class FinallyMethod {
    public static void main(String args[]) {
        try {
            System.out.println("The Value :");
            for(int i=1; i<=3; i++) {
                System.out.println(i);
            }
        } catch(Exception e) {
            e.printStackTrace();
        } finally {
            System.out.println("The finally block always executes..");
        }
    }
}
```


} } }

Example of User defined exception in Java

```
class MyException extends Exception{
    String str1;
    MyException(String str2) {
        str1=str2; }
    public String toString(){
        return ("Output String = "+str1) ;
    }
}
class CustomException{
    public static void main(String args[]){
        try{
            throw new MyException("Custom");
            // I'm throwing user defined custom exception above }
        catch(MyException exp){
            System.out.println("Hi this is my catch block") ;
            System.out.println(exp) ;
        } } }
public class ExceptionSample {
    public static void main(String args[]) throws Exception {
        ExceptionSample es = new ExceptionSample();
        es.displayMymsg(); }
    public void displayMymsg() throws MyException {
        for(int j=8;j>0;j--) {
            System.out.println("j= "+j);
            if(j==7){
                throw new MyException("This is my own Custom Message");
            } } } }
```

Example 1: Arithmetic exception

```
class ExceptionDemo1 {
    public static void main(String args[]) {
        try{
            int num1=30, num2=0;
            int output=num1/num2;
            System.out.println ("Result = " +output); }
        catch(ArithmeticException e){
            System.out.println ("Arithmetic Exception: You can't
            divide an integer by 0");
        } } }
```

Example 2: ArrayIndexOutOfBoundsException Exception

```
class ExceptionDemo2{
    public static void main(String args[]){
        try{
            int a[]=new int[10];
            //Array has only 10 elements
            a[11] = 9; }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println ("ArrayIndexOutOfBoundsException");
        }
    }
}
```

Example 3: NumberFormat Exception

```
class ExceptionDemo3{
    public static void main(String args[]){
        try{
            int num=Integer.parseInt ("XYZ") ;
            System.out.println(num);
        }catch(NumberFormatException e){
            System.out.println("Number format exception occurred");
        }
    }
}
```

Example 4: StringIndexOutOfBounds Exception

```
class ExceptionDemo4{
    public static void main(String args[]){
        try{
            String str="easysteps2buildwebsite";
            System.out.println(str.length());
            char c = str.charAt(0);
            c = str.charAt(40);
            System.out.println(c);
        }catch(StringIndexOutOfBoundsException e){
            System.out.println("StringIndexOutOfBoundsException!!");
        }
    }
}
```

Example 5: NullPointerException

```
package beginnersbook.com;
class Exception2 {
    public static void main(String args[]) {
        try{
            String str=null;
            System.out.println (str.length());
        }catch(NullPointerException e){
            System.out.println("NullPointerException..");
        }
    }
}
```