

**JAVA TRAINING CENTER**

**Java Training Center**  
(No.1 in Training & placement)

# Exception Handling WorkBook

**Master the Content...**

GROWTH UNBOUND

**W.B-4**

**Author**

**Som Prakash Rai**

# **JAVA TRAINING CENTER**

## **Topic**

**Jtc 1: Example using try and catch blocks. (C)**

**Jtc 2: Example using try and multiple catch blocks.**

**Jtc 3: Example using multiple try and catch blocks.**

**Jtc 4: Example using try , catch and finally blocks.**

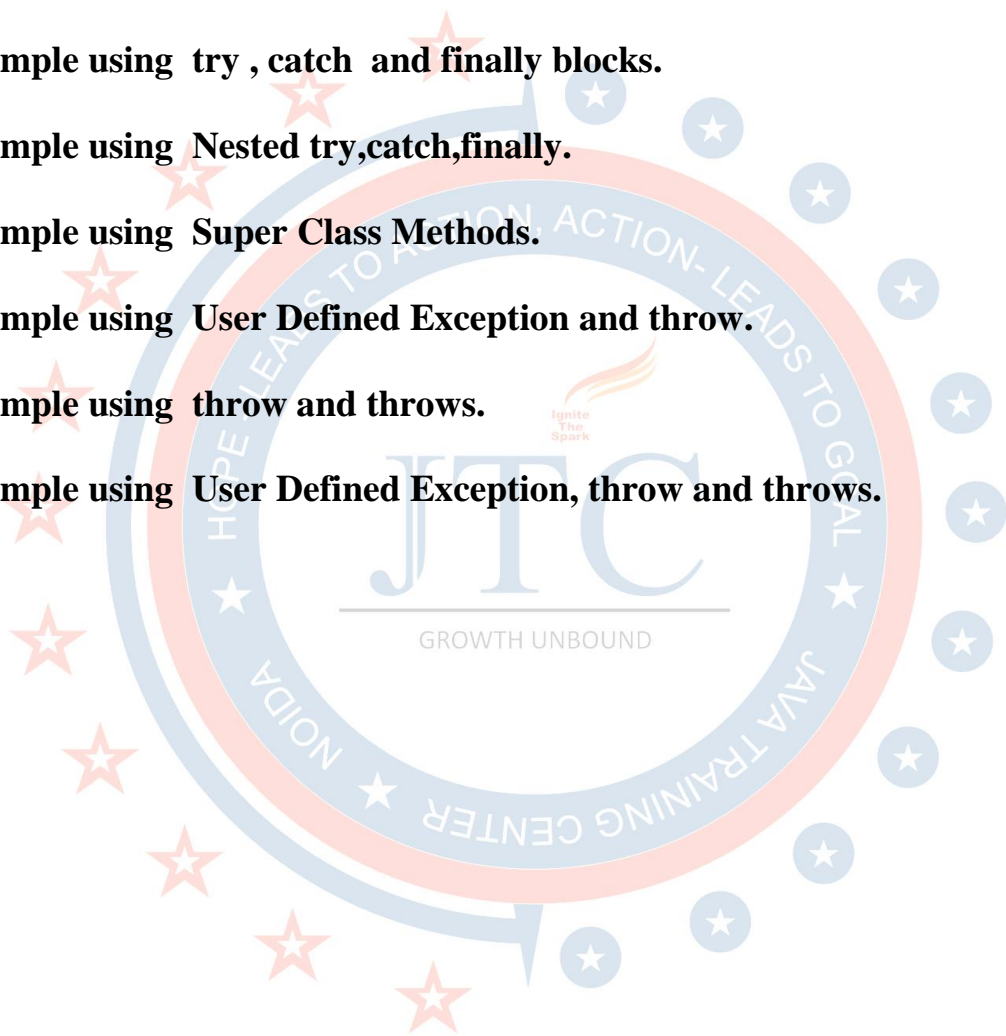
**Jtc 5: Example using Nested try,catch,finally.**

**Jtc 6: Example using Super Class Methods.**

**Jtc 7: Example using User Defined Exception and throw.**

**Jtc 8: Example using throw and throws.**

**Jtc 9: Example using User Defined Exception, throw and throws.**



# JAVA TRAINING CENTER

## Jtc 2: Example using try and multiple catch blocks.

```
public class Jtc2 {
/*
* @Author   : Som Prakash Rai
* @Join     : Java Training Center
* @visit    : www.jtcindia.org
* @Call     : +91-9990399111
* */
public static void main(String[] as) {
System.out.println("main begin");
try{
int x=Integer.parseInt(as[0]);
System.out.println(x);
int y=10/x;
System.out.println(y);
}
//System.out.println("not ok");
catch(ArrayIndexOutOfBoundsException e){
System.out.println("Please Provide the value");
System.out.println(e);
}
//System.out.println("not ok");
catch(NumberFormatException e){
System.out.println("Please Provide integer number");
System.out.println(e);
}
catch(ArithmeticException e){
System.out.println("Please Dont Provide Zero");
System.out.println(e);
}
catch(Exception e){
System.out.println("Please Dont do this");
System.out.println(e);
}
System.out.println("main end");
}
}
```

# JAVA TRAINING CENTER

## Jtc 3: Example using multiple try and catch blocks.

```
public class Jtc3 {  
    /*  
    * @Author   : Som Prakash Rai  
    * @Join     : Java Training Center  
    * @visit    : www.jtcindia.org  
    * @Call     : +91-9990399111  
    * */  
  
    public static void main(String[] as) {  
        System.out.println("main begin");  
        String s=null;  
        try{  
            s=as[0];  
        }  
        catch(ArrayIndexOutOfBoundsException e){  
            System.out.println("Please Provide the value");  
            System.out.println(e);  
        }  
  
        int x=0;  
        try{  
            x=Integer.parseInt(s);  
            System.out.println(x);  
        }  
        catch(NumberFormatException e){  
            System.out.println("Please Provide integer number");  
            System.out.println(e);  
        }  
        try{  
            int y=10/x;  
            System.out.println(y);  
        }  
        catch(ArithmeticException e){  
            System.out.println("Please Dont Provide Zero");  
            System.out.println(e);  
        }  
        System.out.println("main end");  
    }  
}
```

# JAVA TRAINING CENTER

## Jtc 4: Example using try , catch and finally blocks.

```
public class Jtc4 {
/*
 * @Author   : Som Prakash Rai
 * @Join     : Java Training Center
 * @visit    : www.jtcindia.org
 * @Call     : +91-9990399111
 */

public static void main(String[] args) {
System.out.println("IN MAIN ");
int ab = ArithmeticOperation.getRemainder("12", "7");
System.out.println("In Main rem\t:" + ab);
int xy = ArithmeticOperation.getRemainder("12", "0");
System.out.println("In Main rem\t:" + xy);
try {
int rt = ArithmeticOperation.getRemainder("12", "J");
System.out.println("In Main rem\t:" + rt);
} catch (NumberFormatException e) {
System.out.println("== catch in Main Method ==");
}
System.out.println("\n-- Main Completed --");
}

class ArithmeticOperation {
public static int getRemainder(String val1, String val2) {
int rem = -1;
try {
System.out.println("\n** getRemainder **");
System.out.println("Values\t:" + val1 + "\t" + val2);
System.out.println("-- Converting into int type --");
int ab = Integer.parseInt(val1);
int bc = Integer.parseInt(val2);
System.out.println("-- Accessing remainder --");
rem = ab % bc;
System.out.println("Remainder \t:" + rem);
return rem;
} catch (ArithmeticException e) {
System.out.println("-- catch in getRemainder --");
} finally {
System.out.println("++ Finally Block in getRemainder ++");
}
return rem; }}
```

# JAVA TRAINING CENTER

## Jtc 5: Example using Nested try,catch,finally.

### Sample Code

```
try{
    Stmt 1;
    try{
        Stmt 2;
    }catch(Exception e){
        Stmt 3;
    }finally{
        Stmt 4;
    }
    Stmt 5;
}catch(Exception e){
    Stmt 6;
    try{
        Stmt 7;
    }catch(Exception e){
        Stmt 8;
    }finally{
        Stmt 9;
    }
    Stmt 10;
}
finally{
    Stmt 11;
    try{
        Stmt 12;
    }catch(Exception e){
        Stmt 13;
    }finally{
        Stmt 14;
    }
    Stmt 15;
}
```

## Jtc 6: Example using Super Class Methods.

```
import java.io.*;
import java.io.*;
public class Jtc6 {
```



# JAVA TRAINING CENTER

```
/*
 * @Author   : Som Prakash Rai
 * @Join     : Java Training Center
 * @visit    : www.jtcindia.org
 * @Call     : +91-9990399111
 */
public static void main(String[] args) throws Exception {
try{
new A().show();
}catch(Exception e){
System.out.println(e.getClass().getName());
System.out.println(e.getMessage());
System.out.println(e);
System.out.println(e.getCause());
System.out.println();
e.printStackTrace();
e.printStackTrace(new PrintStream(new FileOutputStream("ex.txt")));
}
}
}

class A {
void show() {
System.out.println("-- show in A class --");
new B().m1();
}
}

class B {
void m1() {
System.out.println("-- m1 in B class --");
new C().m2();
}
}

class C {
void m2() {
System.out.println("-- m2 in C class --");
new D().m3();
}
}
```

# JAVA TRAINING CENTER

```
class D {  
void m3() {  
System.out.println("-- m3 in D class --");  
int x=10/0;  
}  
}
```

## Jtc 7: Example using User Defined Exception and throw .

```
public class Jtc7 {  
/*  
* @Author   : Som Prakash Rai  
* @Join     : Java Training Center  
* @visit    : www.jtcindia.org  
* @Call     : +91-9990399111  
* */  
  
public static void main(String[] args) {  
StudentService serv = new StudentService();  
try {  
String nm = serv.getStudentName(-23);  
System.out.println("in Main Name\t:" + nm);  
} catch (Exception e) {  
e.printStackTrace();  
}  
System.out.println();  
try {  
String nm = serv.getStudentName(98);  
System.out.println("in Main Name\t:" + nm);  
} catch (Exception e) {  
System.out.println(e);  
System.out.println(e.getCause());  
System.out.println("----");  
e.printStackTrace();  
}  
System.out.println();  
try {  
String nm = serv.getStudentName(99);  
System.out.println("in Main Name\t:" + nm);  
} catch (Exception e) {  
System.out.println(e);  
System.out.println(e.getCause());  
System.out.println("----");  
}
```



# JAVA TRAINING CENTER

```
e.printStackTrace();
}
System.out.println("\n-- Main Completed --");
}
}
```

```
class StudentService {
String getName(int id) {
String nm = null;
// DB Impl
if (id < 1) {
throw new InvalidIdException(id);
} else if (id == 98) {
StudentNotFoundException e = new StudentNotFoundException();
e.initCause(new IdNotFoundException(id));
throw e;
}
nm = "Somprakash";
return nm;
}
}
```

```
class InvalidIdException extends RuntimeException {
int id;
InvalidIdException(int id) {
this.id = id;
}
public String getMessage() {
return String.valueOf(id);
}
}
```

```
class StudentNotFoundException extends RuntimeException { }
class IdNotFoundException extends RuntimeException {
int id;
IdNotFoundException(int id) {
super(String.valueOf(id));
}
}
```

**Jtc 8: Example using throw and throws.**

```
public class Jtc8{
```

# JAVA TRAINING CENTER

```
/*
 * @Author   : Som Prakash Rai
 * @Join     : Java Training Center
 * @visit    : www.jtcindia.org
 * @Call     : +91-9990399111
 */
public static void main(String[] as) throws IOException,ArithmeticException {
    Hello h=new Hai();
    h.m1(1);
}

class AException extends IOException{ }

class Hello {
    void m1(int x) throws IOException,ArithmeticException{
        if(x==1){
            throw new ArithmeticException();
        }
        else{
            throw new IOException();
        }
    }
}

class Hai extends Hello{
    // OK
    public void m1(int x){
        System.out.println("Hai-m1");
    }
    /* OK
    public void m1(int x) throws IOException,ArithmeticException{
        System.out.println("Hai-m1");
    }
    */
    /* OK
    public void m1(int x) throws AException,ArithmeticException{
        System.out.println("Hai-m1");
    }
    */
    /* NOT OK
    public void m1(int x) throws Exception,ArithmeticException{
```

# JAVA TRAINING CENTER

```
System.out.println("Hai-m1");
}
*/
/* NOT OK
public void m1(int x) throws AException,ArithmeticException,SQLException{
System.out.println("Hai-m1");
}
*/
}
```


## Jtc 9: Example using User Defined Exception ,throw and throws.

```
public class Jtc9{
/*
* @Author   : Som Prakash Rai
* @Join     : Java Training Center
* @visit    : www.jtcindia.org
* @Call     : +91-9990399111
* */
public static void main(String[] args) {
EmployeeService serv = new EmployeeService();
try {
serv.getEmployeeName(null);
} catch (Exception e) {
e.printStackTrace();
}
System.out.println();
try {
serv.getEmployeeName("111");
} catch (Exception e) {
e.printStackTrace();
}
System.out.println();
try {
serv.getEmployeeName("JT-99111");
} catch (Exception e) {
e.printStackTrace();
}
System.out.println();
try {
String nm = serv.getEmployeeName("JTC-222");
System.out.println("Name in Main\t:" + nm);
} catch (Exception e) {
```

# JAVA TRAINING CENTER

```
e.printStackTrace();  
}  
System.out.println("\n-- Main Completed --");  
}  
}
```

```
class EmployeeService {  
    String getEmployeeName(String id) throws InvalidEmployeeIdException {  
        String nm = null;  
        if (id == null)  
            throw new NullPointerException();  
        else if (id.length() != 7)  
            throw new InvalidEmployeeIdException(id);  
        else if (!id.startsWith("JTC"))  
            throw new InvalidEmployeeIdException(id);  
        else  
            nm = "Somprakash";  
        return nm;  
    }  
}  
class InvalidEmployeeIdException extends Exception {  
    InvalidEmployeeIdException(String id) {  
        super(id);  
    }  
}
```



# **JAVA TRAINING CENTER**

## **Exception Handling**

```
public class Throwable extends Object implements Serializable{
    public Throwable();
    public void printStackTrace();
    public void printStackTrace(PrintStream);
    public void printStackTrace(PrintWriter);
    public String getLocalizedMessage();
    public String getMessage();
    public String toString();
    public Throwable(String);
    public Throwable getCause();
    public Throwable(Throwable);
    public Throwable(String,Throwable);
    public synchronized Throwable initCause(Throwable);
}

public class Exception extends Throwable{
    static final long serialVersionUID;
    public Exception();
    public Exception(String);
    public Exception(Throwable);
    public Exception(String,Throwable);
}

public class Error extends Throwable{
    static final long serialVersionUID;
    public Error();
    public Error(String);
    public Error(Throwable);
    public Error(String,Throwable);
```





# **JAVA TRAINING CENTER**

## **List of Exception classes in java.lang package**

<b>ArithmeticException</b>	<b>ArrayIndexOutOfBoundsException</b>
<b>ArrayStoreException</b>	<b>ClassCastException</b>
<b>ClassNotFoundException</b>	<b>CloneNotSupportedException</b>
<b>IllegalAccessException</b>	<b>IllegalArgumentException</b>
<b>IllegalMonitorStateException</b>	<b>IllegalStateException</b>
<b>IllegalThreadStateException</b>	<b>IndexOutOfBoundsException</b>
<b>InstantiationException</b>	<b>InterruptedException</b>
<b>NegativeArraySizeException</b>	<b>NoSuchFieldException</b>
<b>NoSuchMethodException</b>	<b>NullPointerException</b>
<b>NumberFormatException</b>	<b>RuntimeException</b>
<b>StringIndexOutOfBoundsException</b>	<b>UnsupportedOperationException</b>

## **List of Error classes in java.lang package**

<b>AbstractMethodError</b>	<b>AssertionError</b>
<b>ClassCircularityError</b>	<b>ClassFormatError</b>
<b>ExceptionInInitializerError</b>	<b>IllegalAccessError</b>
<b>IncompatibleClassChangeError</b>	<b>InstantiationError</b>
<b>InternalError</b>	<b>NoClassDefFoundError</b>
<b>NoSuchFieldError</b>	<b>NoSuchMethodError</b>
<b>OutOfMemoryError</b>	<b>StackOverflowError</b>
<b>UnknownError</b>	
<b>UnsupportedClassVersionError</b>	<b>VirtualMachineError</b>