Problem without exception handling

**public** **class** TryCatchExample1 {

**public** **static** **void** main(String[] args) {

**int** data=50/0; //may throw exception

        System.out.println("rest of the code");

    }

    }

## Solution by exception handling

**public** **class** TryCatchExample2 {

**public** **static** **void** main(String[] args) {

**try**

        {

**int** data=50/0; //may throw exception

        }

            //handling the exception

**catch**(ArithmeticException e)

        {

            System.out.println(e);

        }

        System.out.println("rest of the code");

    }

}

Example to resolve the exception in a catch block.

**public** **class** TryCatchExample3 {

**public** **static** **void** main(String[] args) {

**int** i=50;

**int** j=0;

**int** data;

**try**

        {

        data=i/j; //may throw exception           }

            // handling the exception

**catch**(Exception e)

        {

             // resolving the exception in catch block

            System.out.println(i/(j+2));

        }

    }

}

Types of Exception Examples

**Arithmetic exception**

|  |
| --- |
| // Java program to demonstrate ArithmeticException  class ArithmeticException\_Demo  {      public static void main(String args[])      {          try {              int a = 30, b = 0;              int c = a/b;  // cannot divide by zero            }          catch(ArithmeticException e) {              System.out.println ("Can't divide a number by 0");          }      }  } |

**StringIndexOutOfBound Exception**

|  |  |
| --- | --- |
| // Java program to demonstrate StringIndexOutOfBoundsException  class StringIndexOutOfBound\_Demo  {      public static void main(String args[])      {          try {              String a = "This is like chipping "; // length is 22              char c = a.charAt(24); // accessing 25th element              System.out.println(c);          }          catch(StringIndexOutOfBoundsException e) {              System.out.println("StringIndexOutOfBoundsException");          }      }  }  **NullPointer Exception**   |  | | --- | | //Java program to demonstrate NullPointerException  class NullPointer\_Demo  {      public static void main(String args[])      {          try {              String a = null; //null value              System.out.println(a.charAt(0));          } catch(NullPointerException e) {              System.out.println("NullPointerException..");          }      }  } |  Java Multi-catch block  1. Example 1   **public** **class** MultipleCatchBlock1 {    **public** **static** **void** main(String[] args) {    **try**{  **int** a[]=**new** **int**[5];  a[5]=30/0;  }  **catch**(ArithmeticException e)  {  System.***out***.println("Arithmetic Exception occurs");  }  **catch**(ArrayIndexOutOfBoundsException e)  {  System.***out***.println("ArrayIndexOutOfBounds Exception occurs");  }  **catch**(Exception e)  {  System.***out***.println("Parent Exception occurs");  }  System.***out***.println("rest of the code");  } } |

**Example 2**

**public** **class** MultipleCatchBlock2 {

**public** **static** **void** main(String[] args) {

**try**{

                String s=**null**;

                System.out.println(s.length());

               }

**catch**(ArithmeticException e)

                  {

                   System.out.println("Arithmetic Exception occurs");

                  }

**catch**(ArrayIndexOutOfBoundsException e)

                  {

                   System.out.println("ArrayIndexOutOfBounds Exception occurs");

                  }

**catch**(Exception e)

                  {

                   System.out.println("Parent Exception occurs");

                  }

               System.out.println("rest of the code");

    }

}

Example 3: Without maintaining the order of exceptions

**class** MultipleCatchBlock3{

**public** **static** **void** main(String args[]){

**try**{

**int** a[]=**new** **int**[5];

    a[5]=30/0;

   }

**catch**(Exception e){System.out.println("common task completed");}

**catch**(ArithmeticException e){System.out.println("task1 is completed");}

**catch**(ArrayIndexOutOfBoundsException e){System.out.println("task 2 completed");}

   System.out.println("rest of the code...");

 }

}

# **ExceptionHandling with MethodOverriding in Java Examples**

Example 1:

**import** java.io.\*;

**class** Parent{

**void** msg(){System.out.println("parent");}

}

**class** TestExceptionChild **extends** Parent{

**void** msg()**throws** IOException{

    System.out.println("TestExceptionChild");

  }

**public** **static** **void** main(String args[]){

   Parent p=**new** TestExceptionChild();

   p.msg();

  }

}

Example 2:

**import** java.io.\*;

**class** Parent{

**void** msg(){System.out.println("parent");}

}

**class** TestExceptionChild1 **extends** Parent{

**void** msg()**throws** ArithmeticException{

    System.out.println("child");

  }

**public** **static** **void** main(String args[]){

   Parent p=**new** TestExceptionChild1();

   p.msg();

  }

}

Example 3:

**import** java.io.\*;

**class** Parent{

**void** msg()**throws** ArithmeticException{System.out.println("parent");}

}

**class** TestExceptionChild2 **extends** Parent{

**void** msg()**throws** Exception{System.out.println("child");}

**public** **static** **void** main(String args[]){

   Parent p=**new** TestExceptionChild2();

**try**{

   p.msg();

   }**catch**(Exception e){}

  }

}