

Exception Handling

A Java exception is an object that describes an exceptional condition that has occurred in a program. Exceptions are the errors occurred at run time. By using Exceptional handling we can make an abnormal program as a normal program. Bu using exception handling we can predict the flow of a program and can take necessary steps for its normal flow.

The super class of all the exceptions is Throwable class. It has two subclass namely Error and Exception classes. Error is the class which we can't avoid those errors occurred at run time.

Exception is the class where we can handle those errors.Exception has two sub classes namely Checked exception and UncheckedException

Checked Exceptions: Those are the subclasses of the Exception class. These are the errors occurred at run time which can not be programmatically rectified.

Unchecked Exception: These are the errors occurred at run time

which can be programmatically rectified.

These are also called RunTimeExceptions. Most of the exceptions are subclasses of RunTimeException class.

There are five key words used in Exception Handling of Java.

They are :

try

catch

finally

throw

throws

try:

A try block is used to encapsulate a piece of code where we may be expecting some run time errors. If this try block finds any run time error, it throws those errors to a suitable catch block.

If there is no run time error in the try block, then the execution of the program is normal.

A try block must always be associated with at least one catch block or a finally block or both.

A try block may have multiple catch blocks, only one of those multiple catch blocks will be executed.

catch :

The error thrown by a try block is caught in this catch block. A catch block is always followed by a try block.

A catch block takes one parameter of type Exception class.

If there are multiple catch blocks for a given try block, the catch block which contains Super class Exception should come after all sub class Exception catch blocks.

finally :

This finally block is used to execute whether an Exception is caught or uncaught.

This is followed by catch blocks or a simple try block. This is mainly used to know the follow of execution of a given program.

throw :

This is used to throw some predefined Exceptions explicitly. It is mainly used to create User Defined Exceptions.

throws :

This is used to throw Exceptions from methods. Most of the checked exceptions use throws in Java programs. Generally this is used while writing IO programs, JDBC programs and other advanced Java Programs like Servlets, RMI etc...

Java's built in Exceptions :

There are many built in Exceptions in Java. These are of type Exception and its sub class Exceptions.

Some of the important built in Exceptions in Exception class are:

ArithmeticException

NumberFormatException

ArrayIndexOutOfBoundsException

IndexOutOfBoundsException

IOException

InterruptedException etc...

User Defined Exceptions :

In addition to built in Exceptions or predefined Exceptions, we can create User Defined Exceptions also depending upon the type of application.

User Defined Exception can be created by extending Exception class.

Question: List out by an example the way to create user defined exception (or) write a program to create user defined exception.

Solution:

Generally user defined exceptions are used in practical applications. All user defined exceptions extend Exception class.

The following MyownException extends exception class.

```
// program to create user defined exception class named  
//MyownException
```

```
//Program solution
```

```
class MyownException extends Exception
```

```
{
```

```
    MyownException ( )
```

```
    {}
```

```
    public String getMessage ( )
```

```
    {
```

```
        return "in sufficient funds";
```

```
    }
```

```
}
```

```
class Exceptiondemo
{
public static void main (String args [ ] )
{
try
    {
        int amount= Integer. parseInt(args [0]) ;
        if (amount < 500)
        {
            throw new MyownException ( );
        }
    }
catch (MyownException m )
{
    System.out. println (m. getMessage ( ) ) ;
}
```

```
}
```

```
}
```

```
}
```

In the above program, the amount is entered from the command line and if the amount is less than 500, it will throw a user defined exception.