

BASICS OF JAVA

OOPS CONCEPTS

STRING HANDLING

EXCEPTION HANDLING

- Introduction to Exceptions
- try and catch block
- try with resource statement
- throw, throws and finally
- User made Exception Subclass
- Method Overriding with Exception Handling
- Chained Exceptions

JAVA MULTITHREADING

ADVANCED TOPICS

COLLECTION FRAMEWORK

JAVA GUI

REFLECTION API

RMI APPLICATION

INNER CLASS

WRAPPER CLASS

FILE HANDLING

LIST

SET

MAP

QUEUE & DEQUE

JDBC

LAYOUT MANAGERS

ADVERTISEMENT

Java Method Overriding with Exception Handling

ADVERTISEMENT

There are few things to remember when overriding a method with exception handling because method of super class may have exception declared. In this scenario, there can be possible two cases: **either method of super class can declare exception or not.**

If super class method declared exception then the method of sub class may declare same exception class, sub exception class or no exception but can not parent of exception class.

For example, if a method of super class declared ArithmeticException then method of subclass may declare ArithmeticException, its subclass or no exception but cannot declare its super/parent class like: Exception class.

In other scenario, if super class method does not declare **any exception, then sub class overridden method cannot declare checked exception but it can declare unchecked exceptions.** Lets see an example.

Method overriding in Subclass with Checked Exception

Checked exception is the exception which is expected or known to occur at compile time hence they must be handled at compile time.

```
import java.io.*;

class Super
{
    void show() {
        System.out.println("parent class");
    }
}

public class Sub extends Super
{
    void show() throws IOException //Compile time error
    {
        System.out.println("parent class");
    }

    public static void main(String[] args)
    {
        Super s=new Sub();
    }
}
```

In the example above, the method **show()** doesn't throw any exception when its declared/defined in the **Super** class, hence its overridden version in the class **Sub** also cannot throw any checked exception.

ADVERTISEMENT

If we try to do so, we will get a **compile time error**.

Method overriding in Subclass with UnChecked Exception

Unchecked Exceptions are the exception which extend the class **RuntimeException** and are thrown as a result of some runtime error.

```
import java.io.*;

class Super
{
    void show() {
        System.out.println("parent class");
    }
}

class Sub extends Super
{
    void show() throws ArrayIndexOutOfBoundsException
    {
        System.out.println("child class");
    }

    public static void main(String[] args)
    {
        Super s = new Sub();
    }
}
```

OUTPUT:

child class

Because **ArrayIndexOutOfBoundsException** is an unchecked exception hence, overridden **show()** method can throw it.

More about Overriden Methods and Exceptions

If Super class method throws an exception, then Subclass overridden method can throw the same exception or no exception, but must not throw parent exception of the exception thrown by Super class method.

It means, if Super class method throws object of **NullPointerException** class, then Subclass method can either throw same exception, or can throw no exception, but it can never throw object of **Exception** class (parent of NullPointerException class).

Example of Subclass overridden method with same Exception

Method of a sub class can declare same exception as declared in the super class. See the below example.

```
import java.io.*;

class Super
{
    void show() throws Exception
    {
        System.out.println("parent class");
    }
}

public class Sub extends Super {
    void show() throws Exception //Correct
    {
        System.out.println("Child class");
    }

    public static void main(String[] args)
    {
        try {
            Super s=new Sub();
            s.show();
        }
        catch(Exception e){}
    }
}
```

OUTPUT:

child class

Example of Subclass overridden method with no Exception

It is optional to declare the exception in sub class during overriding. If method of super class declared an exception then it is upto the subclass to declare exception or not. See the below example.

```
import java.io.*;

class Super
{
    void show() throws Exception
    {
        System.out.println("parent class");
    }
}

public class Sub extends Super {
    void show() //Correct
    {
        System.out.println("Child class");
    }

    public static void main(String[] args)
    {
        try {
            Super s=new Sub();
            s.show();
        }
        catch(Exception e){}
    }
}
```

OUTPUT:

child class

Example of Subclass overridden method with parent Exception

It is not allowed to declare parent class exception in the subclass method, we get compile time error if we try to compile that program. See the below example.

```
import java.io.*;

class Super
{
    void show() throws ArithmeticException
    {
        System.out.println("parent class");
    }
}

public class Sub extends Super {
    void show() throws Exception //Compile time Error
    {
        System.out.println("Child class");
    }

    public static void main(String[] args)
    {
        try {
            Super s=new Sub();
            s.show();
        }
        catch(Exception e){}
    }
}
```

OUTPUT:

Compile time error

← Prev

Next →

ADVERTISEMENT

Java MCQ Tests

Prepare for Java Interview in TCS, Infosys, etc. companies.

Explore

Java Programs

Java programs with code and output for practice.

Explore

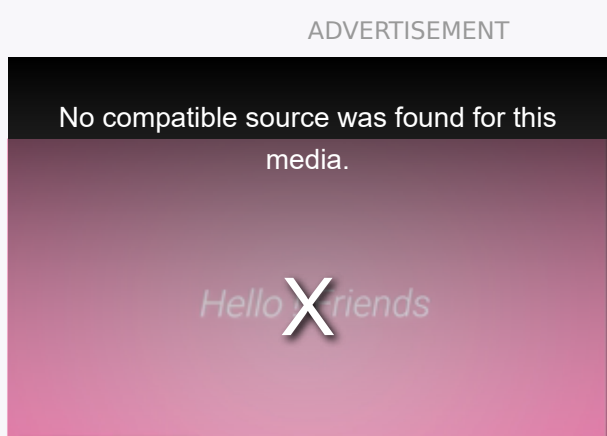
Spring Framework

Learn the most widely used Java framework in the World.

Explore

ADVERTISEMENT

ADVERTISEMENT



ADVERTISEMENT