

Blogs	Core java	PageNo:	<input type="text"/>	<input type="button" value="Go"/>	<input type="button" value=" <<"/>	<input type="button" value=" >>"/>
<input type="text" value="--Tag--"/>	<input type="button" value="Search"/>					

Exception - What is output of

Q: What is output of

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println(test());  
    }  
  
    public static int test() {  
  
        int k = 5;  
        String name = "Vikay"; // Dinanath Chohan";  
        try {  
            System.out.println(name.charAt(6));  
            System.out.println(name.length());  
  
            int i = 15 / k;  
            System.out.println(i);  
        } catch (StringIndexOutOfBoundsException e) {  
  
            System.out.println("-->" + e.getMessage());  
            System.out.println("String abhi choti he");  
            return 10;  
        } catch (RuntimeException e) {  
            System.out.println("Caught by Parent" + e.getMessage());  
            return 15;  
        } finally {  
            System.out.println("This is Finally");  
            return 20;  
        }  
        // return 100;  
    }  
}
```

How do you handle exception in your Application?

Q: How do you handle/propagate exceptions in your Application?

A: We handle three types of exceptions in our application:

1. Database Exception thrown by DAO
2. Business Exception throws by Service classes
3. Input validation exception throws Controller classes

(1) **Database Exceptions** are mainly JDBC Exceptions thrown by DAO classes and handled by Service classes.

When Database Exception received by Service class then Service class rolls back the transaction, converts the Exception into a custom BusinessException and propagates it to the Controller Class.

Action class receives Business Exception and sends Error Message to View to display it on the screen.

(2) **Business Exception** is thrown by Service class and handled by Controller class. For example, UserNotFound is thrown in case of invalid login by LoginCtrl class. Controller class receives Business Exception and sends Error Message to View to display it on the screen.

(3) **Input Validation Exception** is managed by Spring MVC validation Framework.

try-catch inside finally

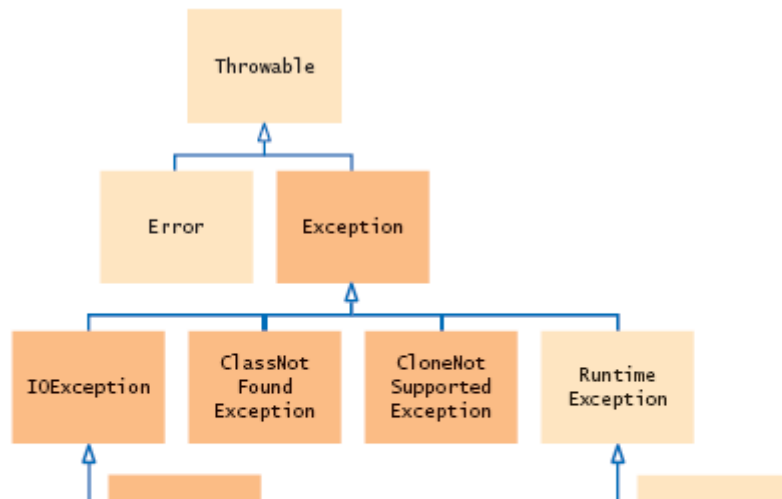
Q : Can you define try-catch block inside finally block?

A: Yes it can be. The finally block made for resource cleanup code that may close a file and database connection. Closing a connection may also raise an exception, in this case we write another try-catch block inside finally block.

```
try{  
  
    ...  
    Connection con = DriverManager.getConnection(...)  
    ... } catch(SQLException ex) {  
    ... } finally {  
        try{con.close() // throws an exception  
        }catch(Exception e){ ... }  
    }  
}
```

What is Exception hierarchy of Java

??



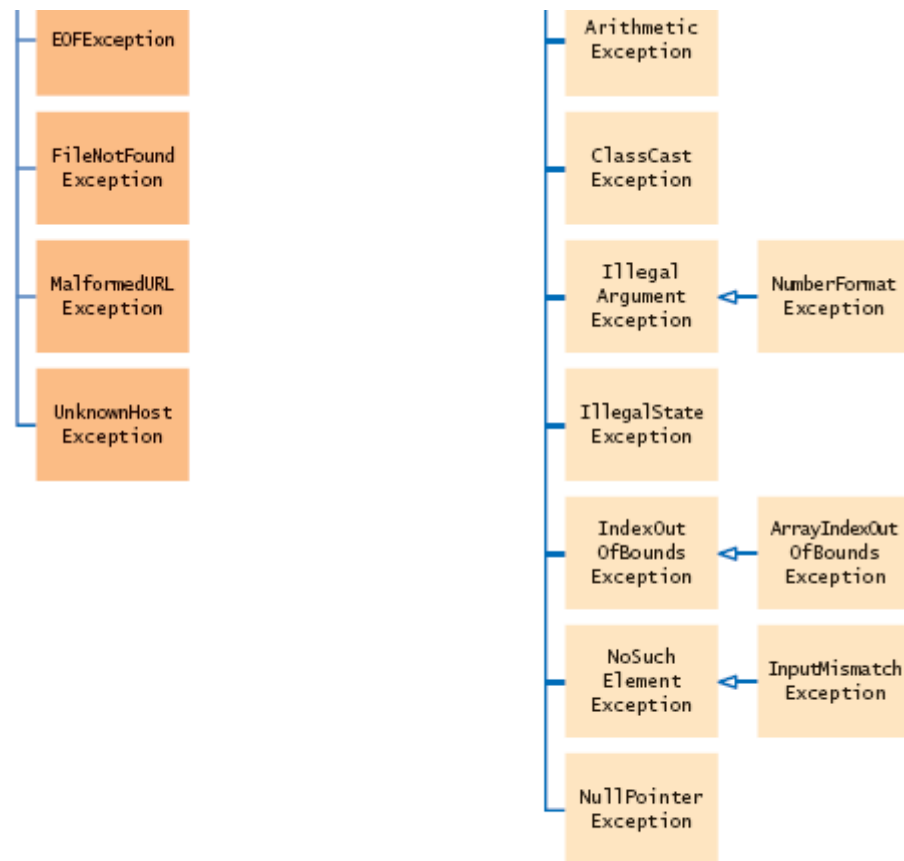


Figure 1 The Hierarchy of Exception Classes

??

Exception - try-catch-finally

Q: How you can stop finally block execution in try-catch?

A: Write `System.exit(0)` in catch block to exit the program;

```
try{  
....  
...  
}catch ( Exception ex){  
....  
    System.exit(0);  
}finally {  
    //I will not execute  
}
```

Q: Can you write return statement in finally block?

A: Yes, but you will get warning message at compile time.

Q: Can you pass integer value other than 0 to `System.exit(n)` method?

A: Yes, but ZERO indicate NORMAL termination of program.