**Java finally return**

I have already discussed about [**finally block**](http://beginnersbook.com/2013/04/java-finally-block/) in my previous tutorial. This post is to learn finally block behavior when a return statement is encountered inside [**try, catch**](http://beginnersbook.com/2013/04/try-catch-in-java/) or finally block.

Consider below example – **What do you think?? Will finally would execute even if there is a return failure statement.**

try {

//try block

...

return success;

}

catch (Exception ex) {

//catch block

.....

return failure;

}

finally {

System.out.println("Inside finally");

}

The answer is yes. **finally** block will execute every time. The only case when it **doesn’t execute** is when it encounters **System.exit().**

**Finally: Example with return statement**

class FinallyDemo

{

public static int myMethod()

{

try {

//try block

return 0;

}

finally {

//finally

System.out.println("Inside Finally block");

}

}

public static void main(String args[])

{

System.out.println(FinallyDemo.myMethod());

}

}

**Output:**

Inside Finally block

0

**Does finally block Overrides the values returned by try-catch block?**

Yes. Finally clause overrides the value returned by try and catch blocks. Consider below example –

public static int myTestingFuncn(){

try{

....

return 5;

} finally {

....

return 19;

}

}

The above code would return value 19 since the value returned by try has been overridden by finally.