If method returns a value and also has try, catch and finally blocks in it, then following two rules need to follow.

1) If finally block returns a value then try and catch blocks may or may not return a value.

2) If finally block does not return a value then both try and catch blocks must return a value.

**public class ReturnValueFromTryCatchFinally**

**{**

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

**{**

**System.out.println(methodReturningValue());**

**}**

**static int methodReturningValue()**

**{**

**try**

**{**

**//This block may or may not return a value as finally block is returning a value**

**}**

**catch (Exception e)**

**{**

**//This block may or may not return a value as finally block is returning a value**

**}**

**finally**

**{**

**return 20;**

**}**

**}**

**}**

**Output: 20**

Example 2:

**public class ReturnValueFromTryCatchFinally**

**{**

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

**{**

**System.out.println(methodReturningValue());**

**}**

**static int methodReturningValue()**

**{**

**try**

**{**

**return 10;**

**}**

**catch (Exception e)**

**{**

**return 20;**

**}**

**finally**

**{**

**//Now, This block may or may not return a value**

**//as both try and catch blocks are returning a value**

**}**

**}**

**}**

Output: 10

**If try-catch-finally blocks are returning a value according to above rules, then you should not keep any statements after finally block. Because they become unreachable and in Java, Unreachable code gives compile time error.**

**public class ReturnValueFromTryCatchFinally**

**{**

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

**{**

**try**

**{**

**return;**

**}**

**catch (Exception e)**

**{**

**return;**

**}**

**finally**

**{**

**return;**

**}**

**System.out.println("Unreachable code");    //Compile Time Error : Unreachable Code**

**}**

**}**

**Finally block overrides any return values from try and catch blocks.**

**public class ReturnValueFromTryCatchFinally**

**{**

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

**{**

**System.out.println(methodReturningValue());    //Output : 50**

**}**

**static int methodReturningValue()**

**{**

**try**

**{**

**return 10;**

**}**

**catch (Exception e)**

**{**

**return 20;**

**}**

**finally**

**{**

**return 50;    //This method returns 50 not 10 or 20**

**}**

**}**

**}**

**Output: 50**

**Finally block will be always executed first even though try and catch blocks are returning the control.**

**public class ReturnValueFromTryCatchFinally**

**{**

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

**{**

**System.out.println(methodReturningValue());    //Output : 10**

**}**

**static int methodReturningValue()**

**{**

**try**

**{**

**return 10;    //control will not be passed to main() method here**

**}**

**catch (Exception e)**

**{**

**return 20;    //Control will not be passed to main() method here**

**}**

**finally**

**{**

**System.out.println("finally block is always executed");**

**//Control will be passed to main() method after executing this block**

**}**

**}**

**}**

**Output:**

finally block is always executed

10