**Try catch finally with return statements Tricky question**:[Ask Question](https://stackoverflow.com/questions/ask)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **This question already has an answer here:**   * [try and Finally giving exception with no return statement , but there is no exception when return statement is written in method](https://stackoverflow.com/questions/25324170/try-and-finally-giving-exception-with-no-return-statement-but-there-is-no-exce) *3 answers*   Trying out a try catch finally use case -  public class Finallyy {  public static void main(String[] args) {  int result = method1();  System.out.println(result);  }  private static int method1() {  int a = 2;  try {  int b = 0;  a = a / a;  System.out.println("try");  return a;  } catch (Exception e) {  System.out.println("caught");  a = 5;  return a;  }  finally {  System.out.println("finally");  a = 10;  return a;  }  }  }  **Output:**  try  finally  10  **If the code is modified to**  finally {  System.out.println("finally");  a = 10;  //return a;  }  **Output:**  try  finally  1  **Question** - Is there some concept of stack where the 'return a' in try block is stored [*if the example is altered, then it applies to try or catch block*], and popped when control leaves the method1() only in the absence of a 'return' in finally block ?  **Update: Solution concluded:**  When the return statement is executed, the value to be returned is stored. When the finally block completes, that value is returned.  The finally block gets always executed as last. So the return in the finally block OVERWRITES the other returns in the try/catch blocks. It's a very bad practice to return or throw an exception from the finally block for this reason.  So to the original question - is it only in the absence of a 'return' in finally block ? NO. It doesn't matter.  **Edit:**  This question is asked on Apr 8 and is answered already. The question that is currently marked along with this as duplicate, is one asked at a later date [August 15]. Hence the new question is to be marked as duplicate, and not this one. However sharing the referenc eto a similar question is good.  [java](https://stackoverflow.com/questions/tagged/java)   |  |  |  | | --- | --- | --- | | [share](https://stackoverflow.com/q/22950240/7489685)[edit](https://stackoverflow.com/posts/22950240/edit) | [edited Aug 22 '14 at 22:06](https://stackoverflow.com/posts/22950240/revisions) | asked Apr 8 '14 at 23:38  [[https://www.gravatar.com/avatar/67b956d69ff2169289c6d1a2d7d36531?s=32&d=identicon&r=PG&f=1](https://stackoverflow.com/users/3250087/spiderman)](https://stackoverflow.com/users/3250087/spiderman)  [spiderman](https://stackoverflow.com/users/3250087/spiderman)  **4,693**52957 | |
| **marked** as duplicate by [Ian Roberts](https://stackoverflow.com/users/592139/ian-roberts)[java](https://stackoverflow.com/badges/tag-badges/java?badgeClass=Gold) Aug 21 '14 at 23:12  This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](https://stackoverflow.com/questions/ask). | |
|  | * Yes. Your "some concept of stack" is the stack. – [Dawood ibn Kareem](https://stackoverflow.com/users/1081110/dawood-ibn-kareem" \o "49,319 reputation) [Apr 8 '14 at 23:40](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35038820_22950240) * @DavidWallace, you mean the value of a is stored in the stack? I mean the value is stored in stack memory as any other values inside a function is stored? – [spiderman](https://stackoverflow.com/users/3250087/spiderman" \o "4,693 reputation) [Apr 9 '14 at 0:00](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039198_22950240) * 1   Yes. The method that you've called puts its return value on the stack. The calling method pops it off. – [Dawood ibn Kareem](https://stackoverflow.com/users/1081110/dawood-ibn-kareem" \o "49,319 reputation) [Apr 9 '14 at 0:13](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039413_22950240)   * @DavidWallace , Although this is not related to my original question... you said "The calling method pops it off". Is it the calling method, or .. the current method, before actually returning the value to the calling function, pops it off the stack and return the value to the calling function? – [spiderman](https://stackoverflow.com/users/3250087/spiderman" \o "4,693 reputation) [Apr 9 '14 at 0:24](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039601_22950240) * 1   The mechanism for returning a value from any method is that the callee pushes the value to the stack, and the caller pops it off. – [Dawood ibn Kareem](https://stackoverflow.com/users/1081110/dawood-ibn-kareem" \o "49,319 reputation) [Apr 9 '14 at 1:47](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35040878_22950240)  Top of Form  Bottom of Form  add a comment |

3 Answers

[active](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return?answertab=votes#tab-top)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote2down voteaccepted | A try block is executed before its finally block.  When the return statement is executed, the value to be returned is stored. When the finallyblock completes, that value is returned.  Note that a is not a value. a is a variable that stores a value. If you change a, you change a, you don't change the value that was stored for the return.  For the technical reason, we go to the [Java Language Specification on the try-finally block](http://docs.oracle.com/javase/specs/jls/se8/html/jls-14.html#jls-14.20.2)  If execution of the try block completes abruptly for any other reason R, then the finally block is executed, and then there is a choice:   * If the finally block completes normally, then the try statement completes abruptly for reason R. * If the finally block completes abruptly for reason S, then the try statement completes abruptly for reason S (**and reason R is discarded**).   [A return statement completes **abruptly**](http://docs.oracle.com/javase/specs/jls/se8/html/jls-14.html#jls-14.21), so if a try with a finally has a return and the finally also has a return, the finally's return supersedes the try's.   |  |  |  | | --- | --- | --- | | [share](https://stackoverflow.com/a/22950271/7489685)[edit](https://stackoverflow.com/posts/22950271/edit) | [edited Apr 8 '14 at 23:46](https://stackoverflow.com/posts/22950271/revisions) | answered Apr 8 '14 at 23:41  [[https://i.stack.imgur.com/5t1BN.png?s=32&g=1](https://stackoverflow.com/users/438154/sotirios-delimanolis)](https://stackoverflow.com/users/438154/sotirios-delimanolis)  [Sotirios Delimanolis](https://stackoverflow.com/users/438154/sotirios-delimanolis)  **188k**37401505 | |
|  | * I am trying to put these into my head! thanks! for all those details +1. – [spiderman](https://stackoverflow.com/users/3250087/spiderman" \o "4,693 reputation) [Apr 8 '14 at 23:58](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039170_22950271) * Does all the return statement considered to be completes abruptly , or only the return statement in a finally block is only considered to be completes abruptly? – [spiderman](https://stackoverflow.com/users/3250087/spiderman" \o "4,693 reputation) [Apr 9 '14 at 0:11](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039392_22950271) * @prash All return, break, throw, and continue statements do not complete normally, ie. they complete abruptly. – [Sotirios Delimanolis](https://stackoverflow.com/users/438154/sotirios-delimanolis" \o "187,664 reputation) [Apr 9 '14 at 0:12](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039409_22950271) * 1   ... which of course is normal. – [Dawood ibn Kareem](https://stackoverflow.com/users/1081110/dawood-ibn-kareem" \o "49,319 reputation) [Apr 9 '14 at 0:13](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039425_22950271)   * 1   Right, it's just terminology used in the specification. – [Sotirios Delimanolis](https://stackoverflow.com/users/438154/sotirios-delimanolis" \o "187,664 reputation) [Apr 9 '14 at 0:14](https://stackoverflow.com/questions/22950240/java-try-catch-finally-return#comment35039445_22950271)  Top of Form  Bottom of Form  add a comment |
| up vote0down vote | A try..catch..finally is actually compiled as a nested try..catch (of the exception you specify) within a try..catch (of "any"), and the "finally" block is cut & pasted by the compiler to the "end" of each code path. (I put "end" in quotes because I can't remember off the top of my head how it handles multiple exits from a "try" or "catch" block.) You can use the javap command (dis-assembler) to experiment on what byte code is generated by your version of Java to correspond to different try..catch..finally combinations.   |  |  | | --- | --- | | [share](https://stackoverflow.com/a/22950607/7489685)[edit](https://stackoverflow.com/posts/22950607/edit) | answered Apr 9 '14 at 0:19  [[https://www.gravatar.com/avatar/cce3699e56107addf58dd200a63c65b3?s=32&d=identicon&r=PG](https://stackoverflow.com/users/1736287/cpurdy)](https://stackoverflow.com/users/1736287/cpurdy)  [cpurdy](https://stackoverflow.com/users/1736287/cpurdy)  **945**410 | |
|  | add a comment |

|  |  |
| --- | --- |
| up vote1down vote | The finally block gets always executed as last. So the return in the finally block OVERWRITES the other returns in the try/catch blocks. It's a very bad practice to return or throw an exception from the finally block for this reason.  Nothing to do with stacks: it's a matter of the fact that the finally block gets always executed as last block, so what is returned or thrown in the finally block erases and overwrites any previous exit value (returned object or exception thrown).  Try this:  try {  throw new NullPointerException();  }  finally {  return 0;  }  The return in the finally block will overwrite (swallow) the NullPointerException... |