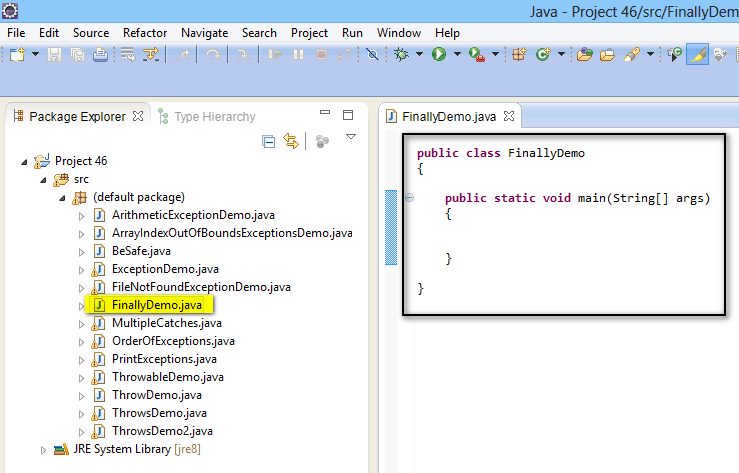
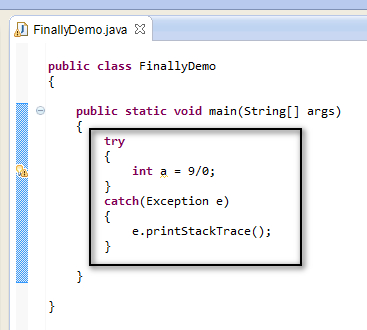
 Using finally in exception handling

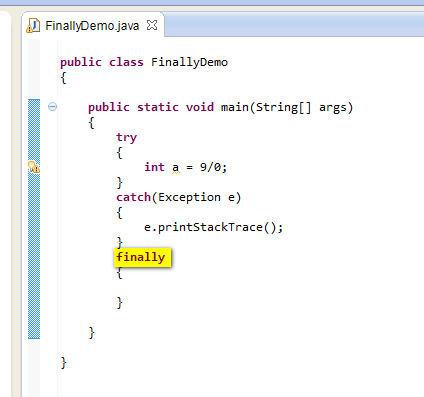
***finally***keyword is used in a try catch block statement to specify a block of code which gets executed regardless of whether the exception is thrown or not thrown.  
  
Lets implement this on Eclipse IDE -  
  
1. Launch Eclipse IDE, create a new java class 'FinallyDemo.java' with main( ) method in the existing project as shown below -

[](https://1.bp.blogspot.com/-b3icfkhenh4/U_L8ypD5cmI/AAAAAAAAb9o/ZEUQB7lscOo/s1600/1.jpg)

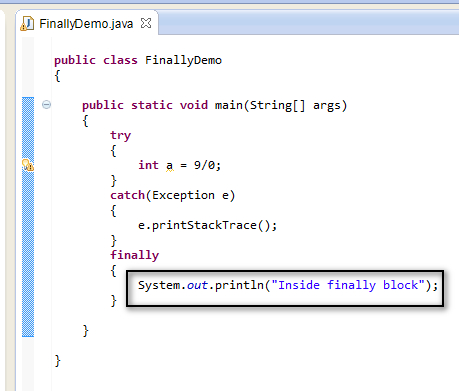
2. Create a try catch block to handle any exception using Exception Class as shown below -

[](https://3.bp.blogspot.com/-uMmaNzYHR3A/U_L9XUk4LaI/AAAAAAAAb9w/G9UhKLRZGyY/s1600/2.jpg)

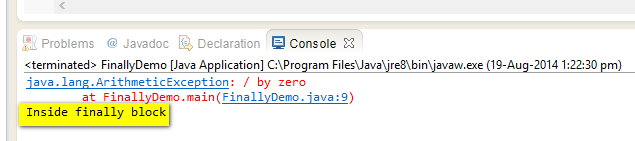
3. Now add ***finally***block to the try catch blocks as shown below -

[](https://4.bp.blogspot.com/-SglM6G8z2BQ/U_L90Txm1cI/AAAAAAAAb94/jssgBjkgef4/s1600/3.jpg)

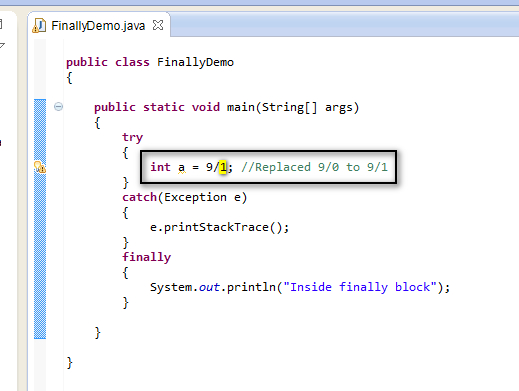
4. Write statements which you want to get executed irrespective of whether the exeception is thrown or not thrown as shown below -

[](https://2.bp.blogspot.com/-mSGGFZCfFBQ/U_MBrJxYdjI/AAAAAAAAb-A/rPNy9A6nFP8/s1600/4.jpg)

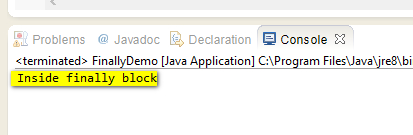
5. Save and Run the Java class file 'FinallyDemo.java' and observe that the print statement inside the ***finally***block got executed, even after the exception is thrown from try block and handled by the catch block as shown below -

[](https://3.bp.blogspot.com/-nNLp_AA-jjc/U_MCTDkt2GI/AAAAAAAAb-I/-hN7jsv-RbM/s1600/5.jpg)

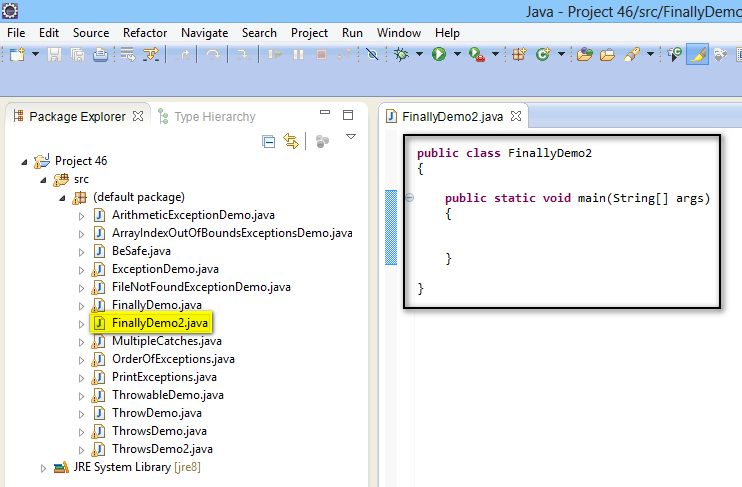
6. Now change the statement in the try block, such that it wont throw any exception as shown below -

[](https://2.bp.blogspot.com/-87ouSTpEh24/U_MCqYlETmI/AAAAAAAAb-Q/gdbdrVWTg5s/s1600/6.jpg)

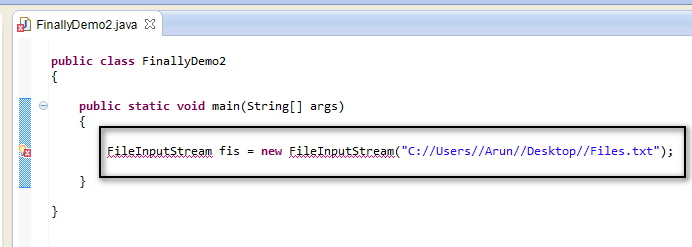
7. Save and Run the Java class file 'FinallyDemo.java' and observe that the print statement inside the ***finally***block got executed, even after the exception is not thrown from try block as shown below -

[](https://3.bp.blogspot.com/-a23V32uLDNo/U_MC3gRLw0I/AAAAAAAAb-Y/j9BlPU9kIG8/s1600/7.jpg)

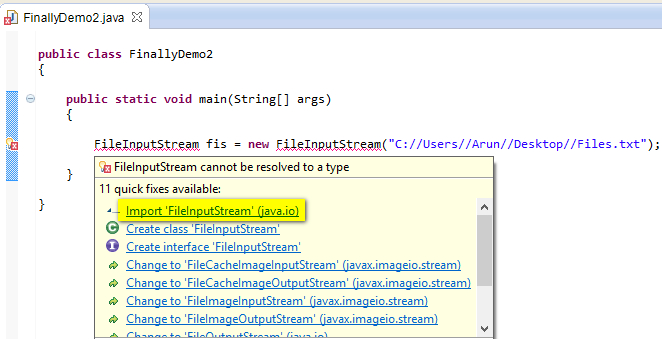
Hence code in the ***finally***block will be executed irrespective of the try catch blocks throwing or handling exceptions.  
  
The main usage of the finally block is to do clean up job.  Keeping cleanup code in a finally block code is always a good practice, even when no exceptions are occurred. The runtime system always executes the code with in the finally block regardless of what happens in the try block. So it is ideal place to clean up code.  
  
Lets take an example of ***FileInputStream***Class. When you create an object for ***FileInputStream***Class, the object refers to a resource file. After using the resource in the program code, we must clean up the resource explicitly at the end by calling the ***close( )*** method. But when we place our code in try catch blocks, we have to add ***finally***block at the end to clean up the resources.  
  
  
Lets implement this on Eclipse IDE -  
  
1. Launch Eclipse IDE, create a new Java Class 'FinallyDemo2.java' with main( ) method in the existing Java Project 'Project 46' as shown below -

[](https://2.bp.blogspot.com/-jcVUGt5QYb4/U_MIjvclqrI/AAAAAAAAb-k/QQeJdQSXKIg/s1600/8.jpg)

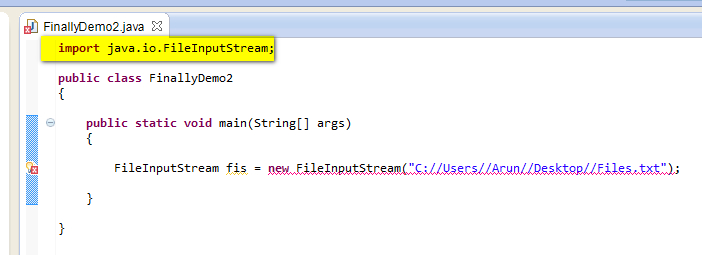
2. Create an object for ***FileInputStream***Class by providing a valid path of a file as shown below -

[](https://1.bp.blogspot.com/-dq_Bc9FVpA4/U_MJzpsB8YI/AAAAAAAAb-s/91GYSdtduaI/s1600/9.jpg)

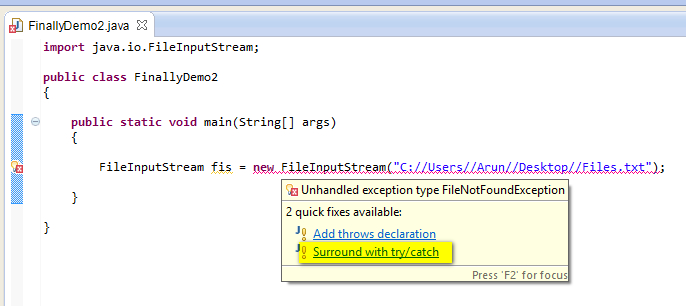
3. View the error and select 'Import FileInputStream (java.io) ' option from the error message as shown below -

[](https://4.bp.blogspot.com/-3EKyk74NEE0/U_MO0BSaRjI/AAAAAAAAb-4/LktN8JchBy8/s1600/10.jpg)

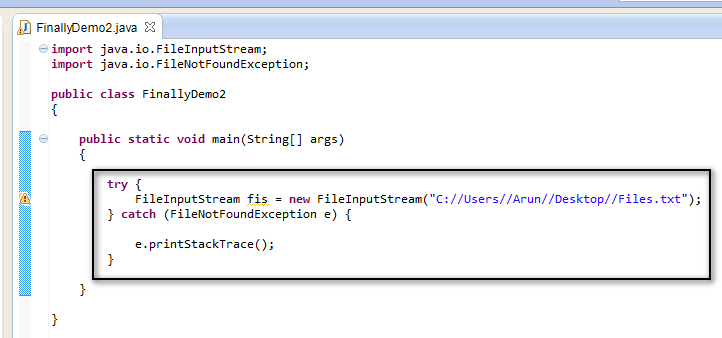
4. Observe that the import statement got added and the error got resolved as shown below -

[](https://4.bp.blogspot.com/-8iqkZYcwBnw/U_MPFGWC94I/AAAAAAAAb_A/DQ09ZcLKdFI/s1600/11.jpg)

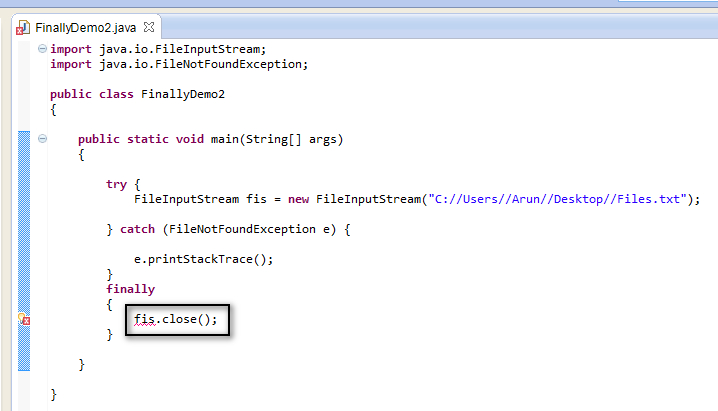
5. View the error displayed after resolving the above error and select 'Surround with try/catch' option as shown below -

[](https://2.bp.blogspot.com/-VTpqy84jCbo/U_MPWdbvX2I/AAAAAAAAb_I/Nz164NnrpVc/s1600/12.jpg)

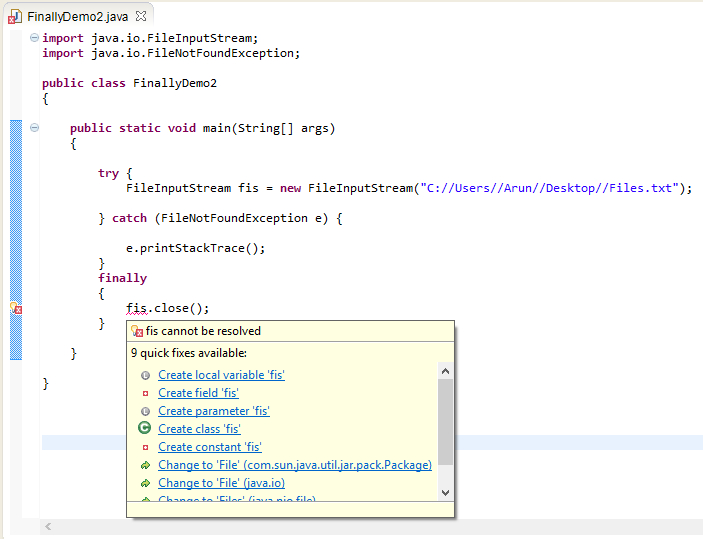
6. ***FileInputStream*** object creation statement got surrounded with try catch blocks and the error got resolved as shown below -

[](https://2.bp.blogspot.com/-kzhKz_5nmVM/U_MPv5rWoII/AAAAAAAAb_Q/epnGXIjY1GY/s1600/13.jpg)

7. Now add a ***finally***block to above try catch block to clean up the resource created by ***FileInputStream***Class using the close( ) method as shown below -

[](https://1.bp.blogspot.com/-3NIpJfPjfxY/U_MRS1j5aTI/AAAAAAAAb_Y/aBV0oZ6TUGo/s1600/14.jpg)

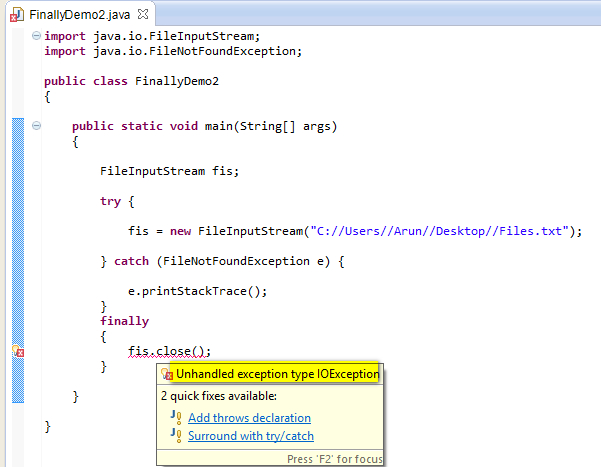
8. View the error as shown below and observe that the object 'fis' of ***FileInputStream***Class is not identified in the finally block as shown below -

[](https://2.bp.blogspot.com/-giqbTj13lB0/U_MRxv4kWpI/AAAAAAAAb_g/pJ6NG8ktjhM/s1600/15.jpg)

9. Now make the FileInputStream object 'fis' recognized by ***finally***block by moving the declaration part of the 'fis' object creation statement from the try block, outside the blocks as shown below -

[](https://2.bp.blogspot.com/-PNUUbWN98nw/U_MSjbs64UI/AAAAAAAAb_o/lWkj6zQL_1k/s1600/16.jpg)

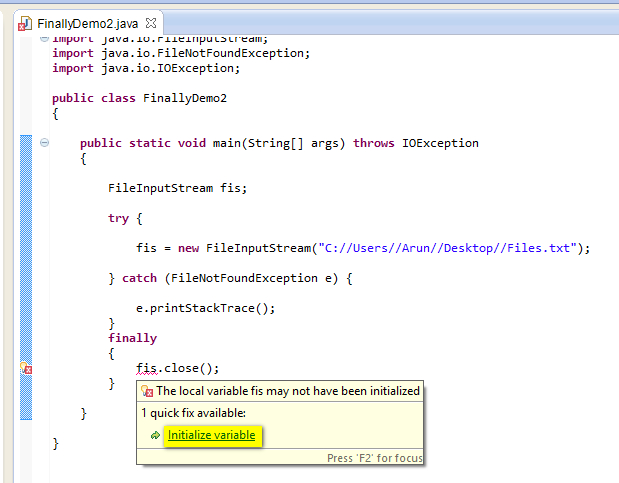
10. View the compiler error that is displayed after resolving the above error and select 'Add throws declaration' option from the error message to resolve the error as shown below -

[](https://3.bp.blogspot.com/-3Rh7M2C6P-o/U_MWlEX4VQI/AAAAAAAAb_w/isbrN_5ZirI/s1600/17.jpg)

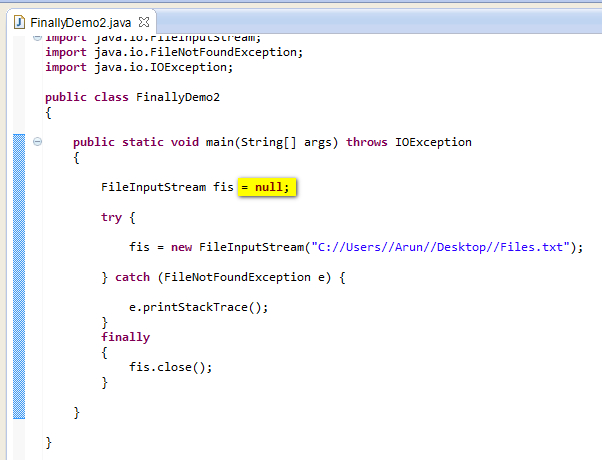
11. Observe that the ***throws***keyword listing ***IOException***got added to declaration part of the ***main( )***method and the error got resolved as shown below***-***

[](https://4.bp.blogspot.com/-AYAWZmc03dw/U_MX2HqCTeI/AAAAAAAAb_4/RoGciZREPh4/s1600/18.jpg)

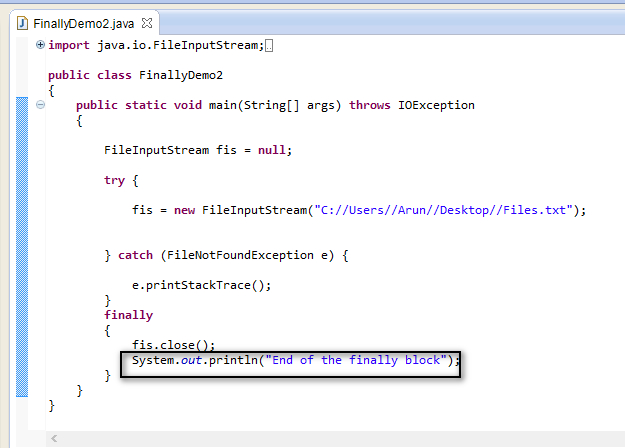
12. Now view the other error displayed after resolving the above error as shown below and select 'Initialize variable' option from the error message as shown below -

[](https://2.bp.blogspot.com/-T0P2OnH1LHU/U_MYLnJtK6I/AAAAAAAAcAA/gA4sz3yT3_4/s1600/19.jpg)

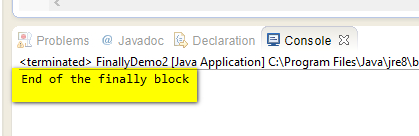
13. Observe that ***fis***object got initialized with null parameter and the error got resolved as shown below -

[](https://3.bp.blogspot.com/-t6ZVxHrl02I/U_MYia5WtoI/AAAAAAAAcAI/_CASIFrz-7c/s1600/20.jpg)

This is the proper way of cleaning up the data by write the statements in the ***finally***block.  
  
14. Now lets add a statement inside the finally block to print the whether the finally block is executed as shown below -

[](https://4.bp.blogspot.com/-zcjWWs_3CTI/U_MaHpmnvjI/AAAAAAAAcAQ/MEabG7xcCmo/s1600/21.jpg)

15. Save and Run the Java Class file 'FinallyDemo2.java' and observe that the statements in the finally block got executed and the result is printed in the output as shown below -

[](https://4.bp.blogspot.com/-dC80zWa2D8U/U_MbZFpQDvI/AAAAAAAAcAY/LnZ71-1f38c/s1600/22.jpg)

Hence the statements in the finally block got executed and using the above example we have used ***finally***block to close the resources by using ***close( )***method.