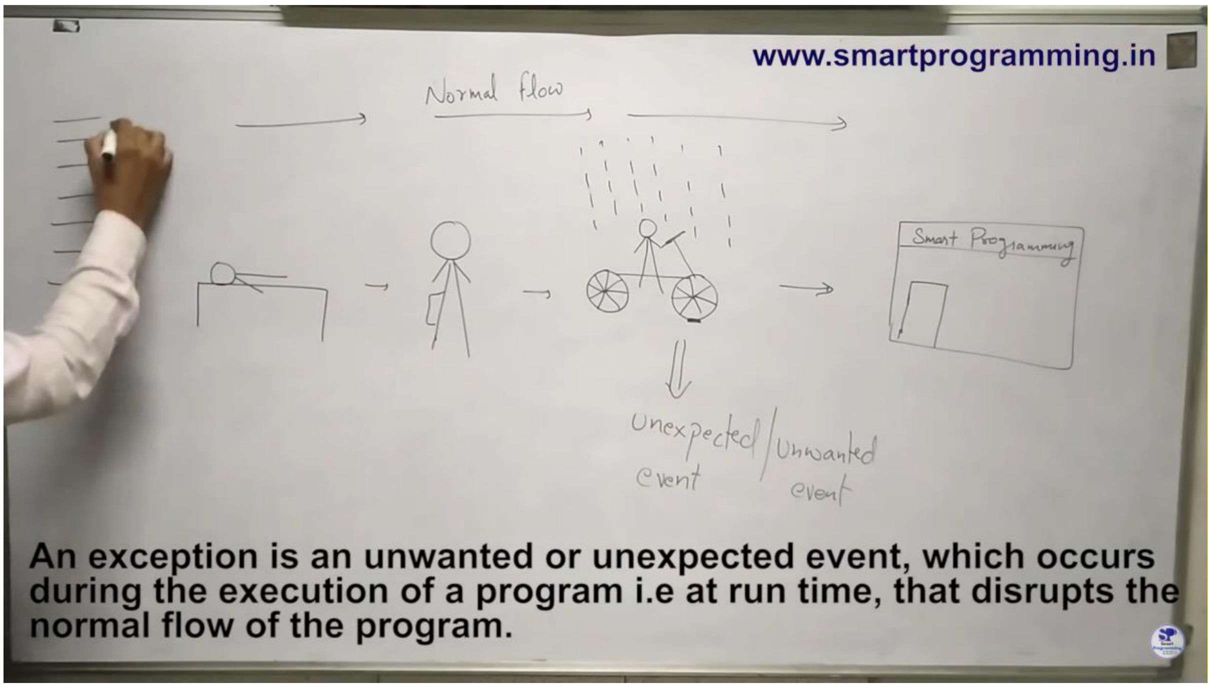
1) What is Exception?

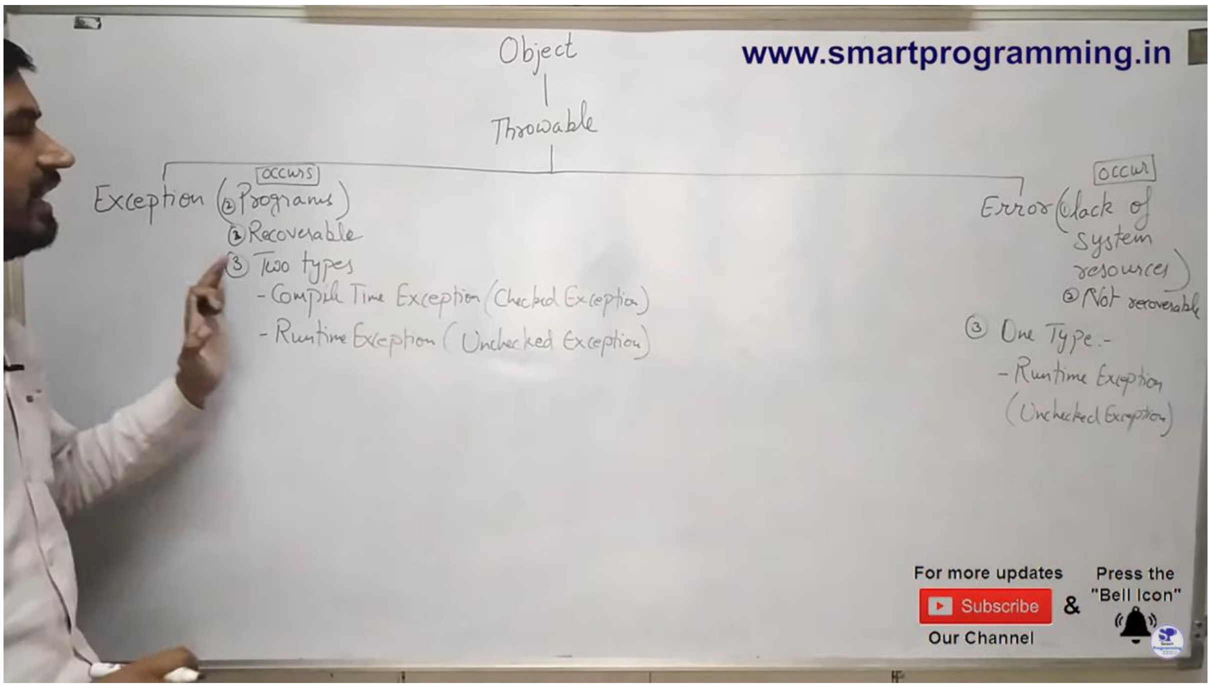


An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program.

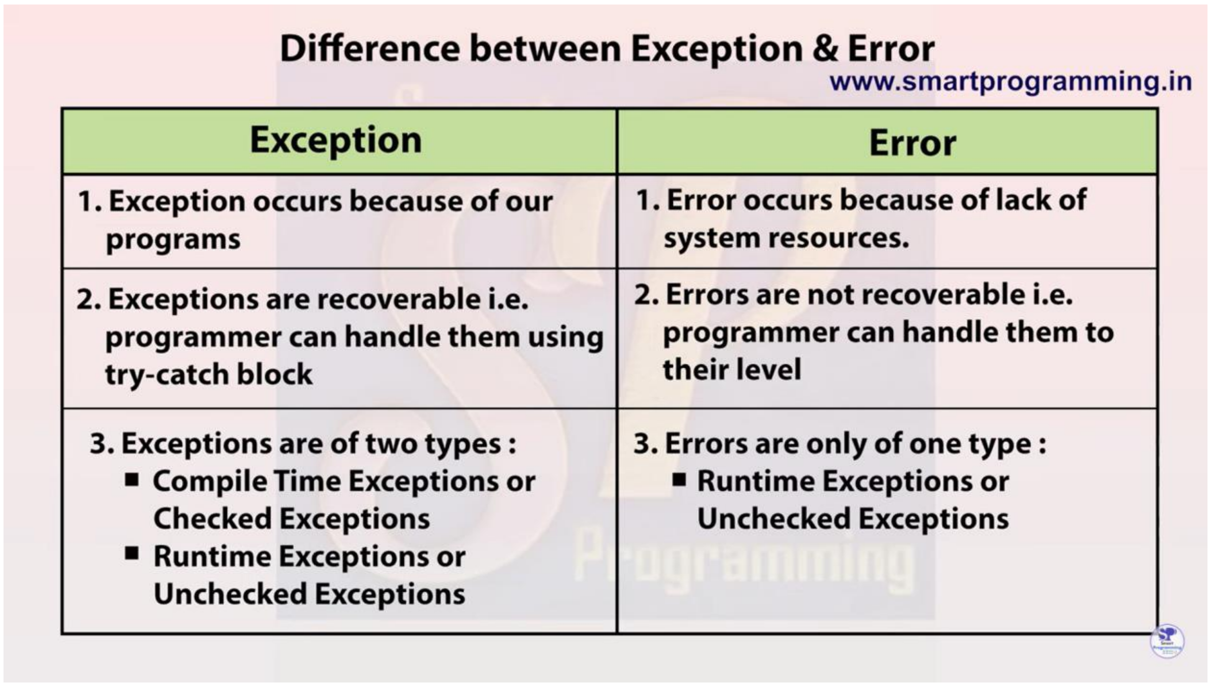
2) Object Class is the Parent Class Of all the classes in Java.

Object is the parent class of all the classes in Java

3) Exception Hierarchy


Object
Throwable
Exception (① Programs) occurs
② Recoverable
③ Two types
- Compile Time Exception (Checked Exception)
- Runtime Exception (Unchecked Exception)
Error (① lack of system resources) occur
② Not recoverable
③ One Type :-
- Runtime Exception (Unchecked Exception)
www.smartprogramming.in
For more updates
Subscribe
Our Channel
&
Press the "Bell Icon"

4) Difference Between Exception & Error.

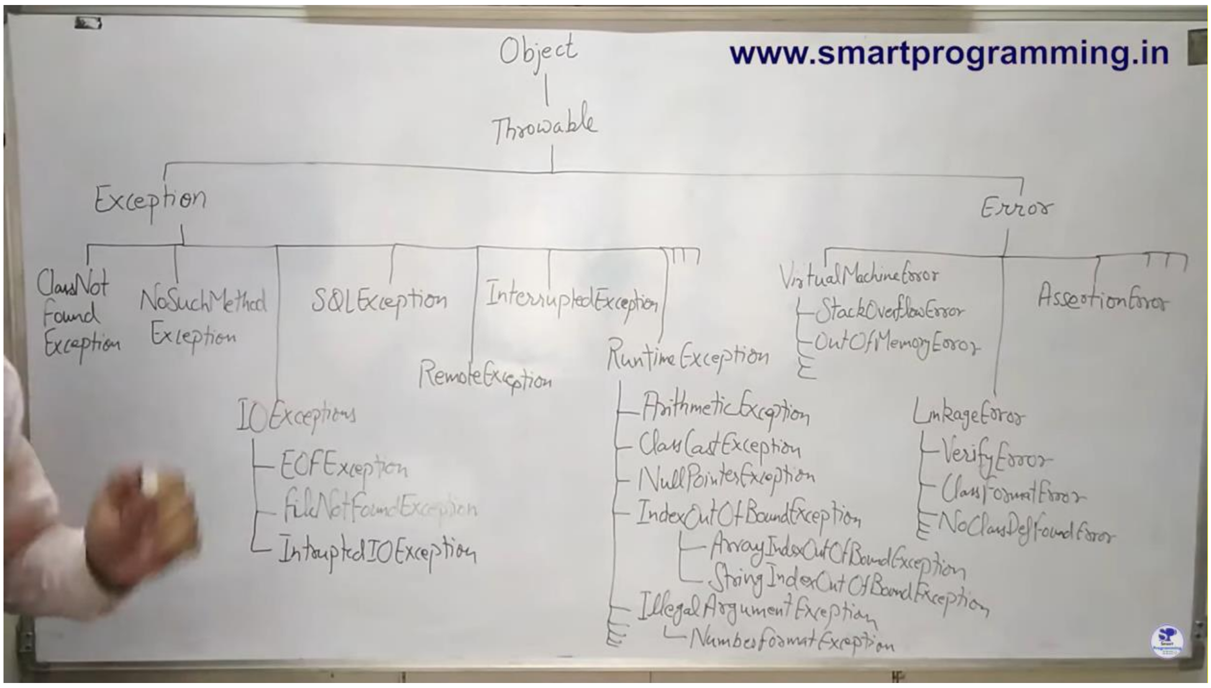


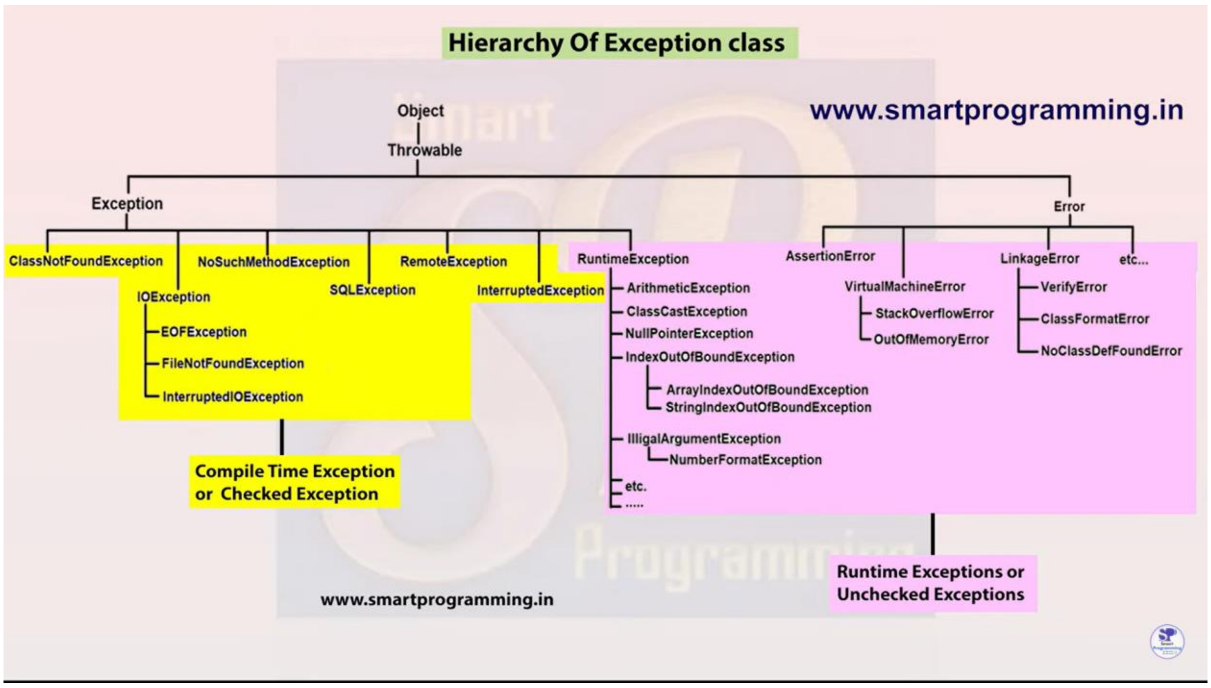
Difference between Exception & Error

www.smartprogramming.in

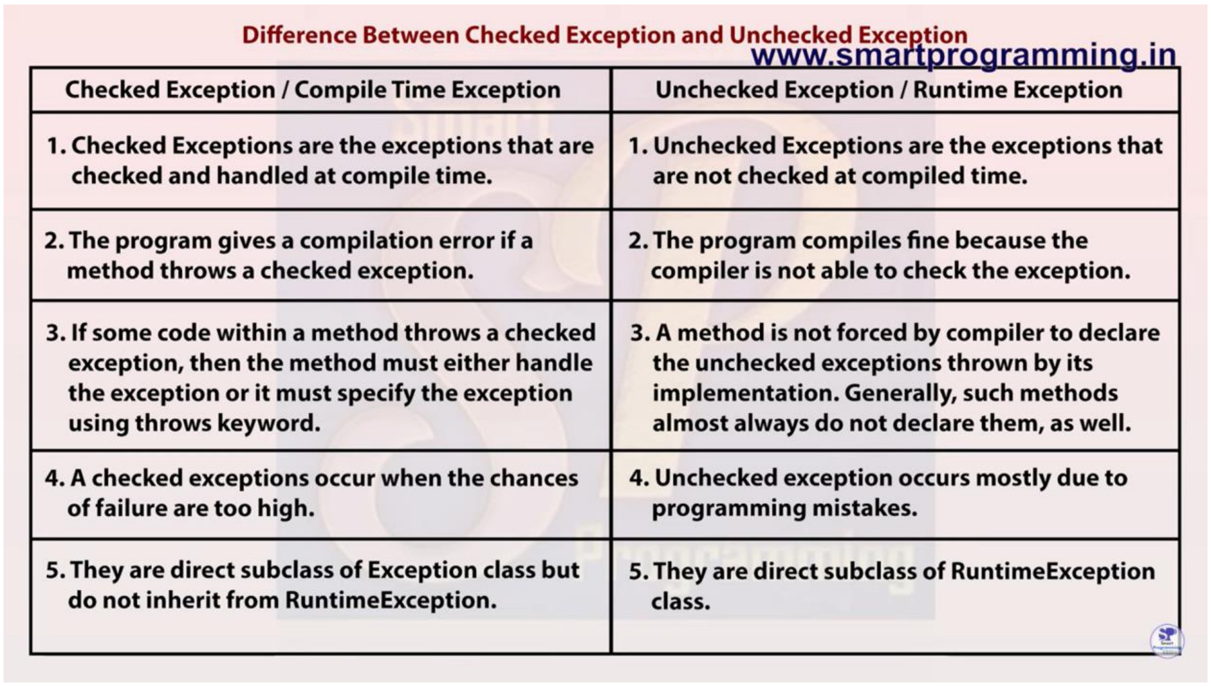
Exception	Error
1. Exception occurs because of our programs	1. Error occurs because of lack of system resources.
2. Exceptions are recoverable i.e. programmer can handle them using try-catch block	2. Errors are not recoverable i.e. programmer can handle them to their level
3. Exceptions are of two types : ▪ Compile Time Exceptions or Checked Exceptions ▪ Runtime Exceptions or Unchecked Exceptions	3. Errors are only of one type : ▪ Runtime Exceptions or Unchecked Exceptions

5) 1000s Of Exceptions available Like this:-





6) Diff Betn Checked & Unchecked Exception



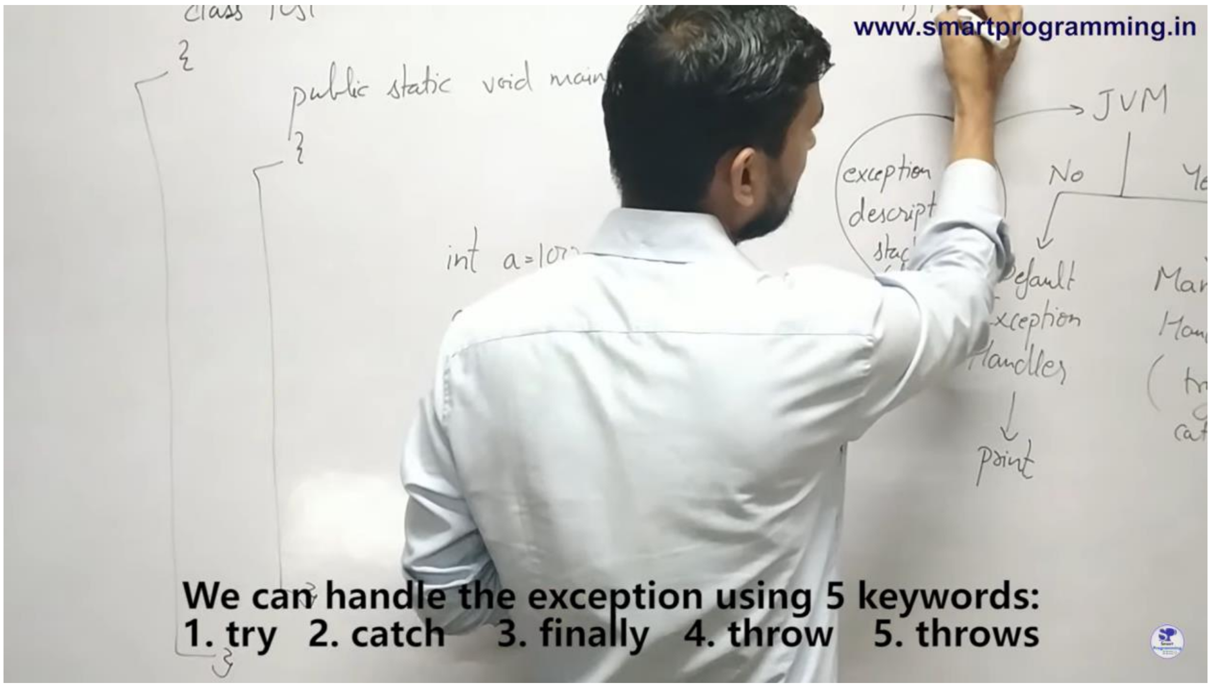
Difference Between Checked Exception and Unchecked Exception

www.smartprogramming.in

Checked Exception / Compile Time Exception	Unchecked Exception / Runtime Exception
1. Checked Exceptions are the exceptions that are checked and handled at compile time.	1. Unchecked Exceptions are the exceptions that are not checked at compiled time.
2. The program gives a compilation error if a method throws a checked exception.	2. The program compiles fine because the compiler is not able to check the exception.
3. If some code within a method throws a checked exception, then the method must either handle the exception or it must specify the exception using throws keyword.	3. A method is not forced by compiler to declare the unchecked exceptions thrown by its implementation. Generally, such methods almost always do not declare them, as well.
4. A checked exceptions occur when the chances of failure are too high.	4. Unchecked exception occurs mostly due to programming mistakes.
5. They are direct subclass of Exception class but do not inherit from RuntimeException.	5. They are direct subclass of RuntimeException class.

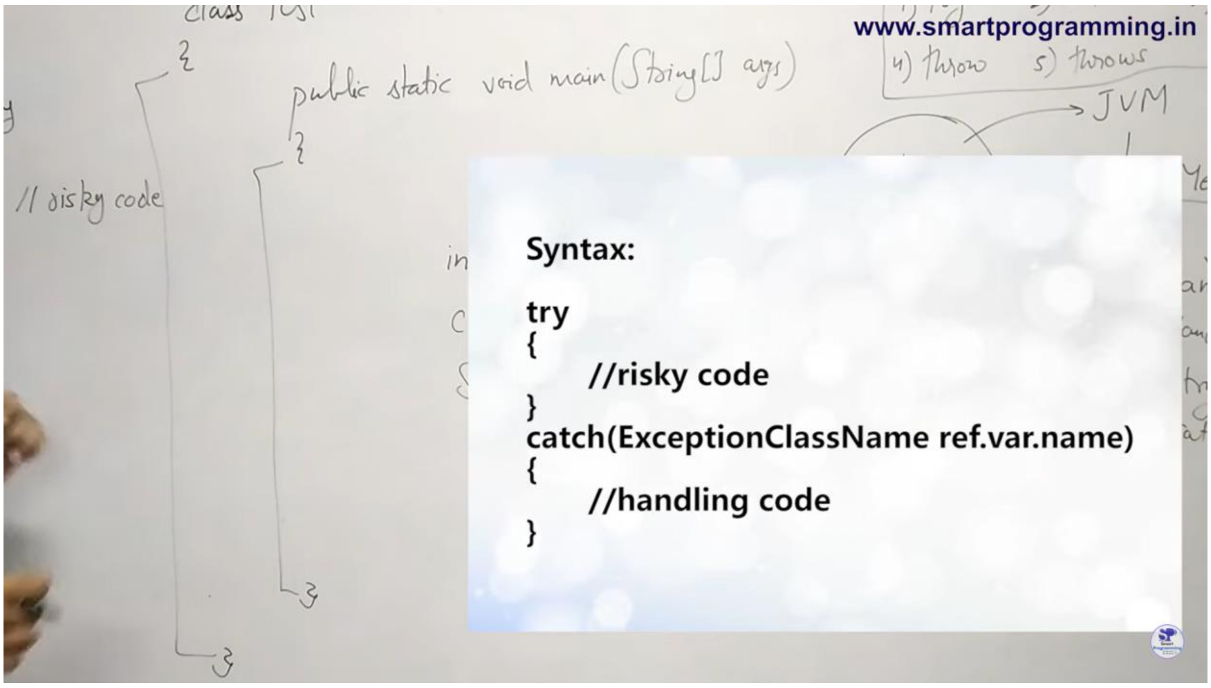
7) How to handle Exceptions?

This are 5 Keywords:-



We can handle the exception using 5 keywords:
1. try 2. catch 3. finally 4. throw 5. throws

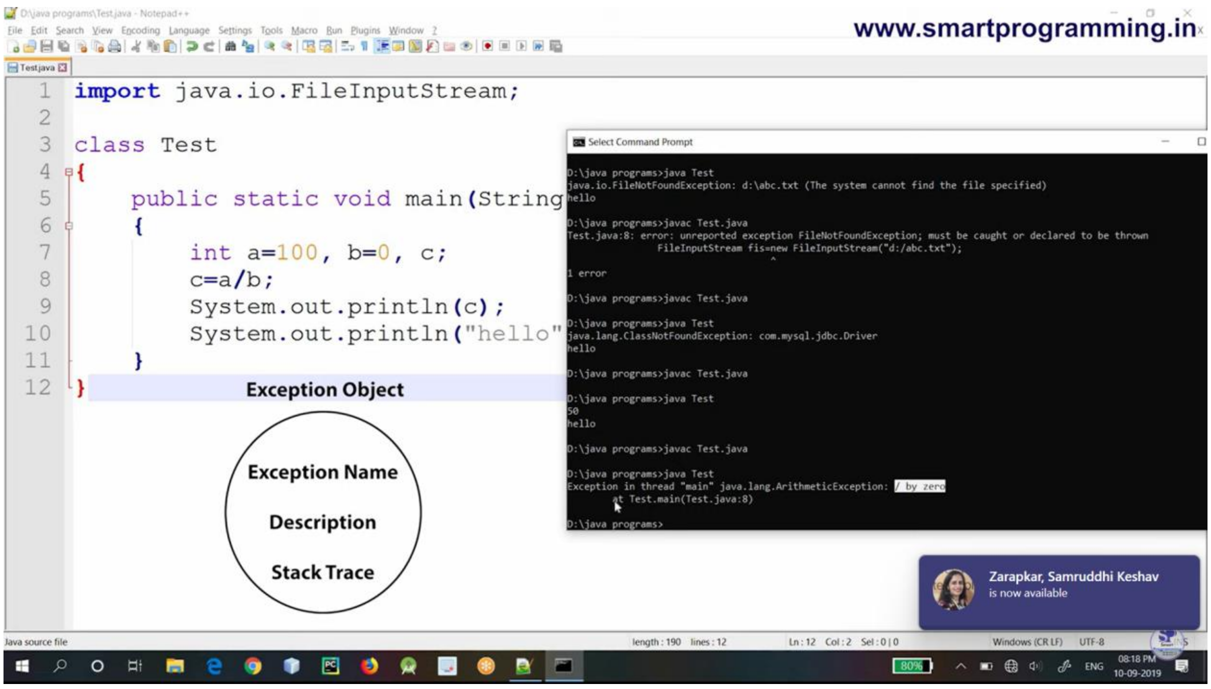
Syntax of Try Catch:-



Syntax:

```
try
{
    //risky code
}
catch(ExceptionClassName ref.var.name)
{
    //handling code
}
```

Exception Object prints the following Details:-



```java
import java.io.FileInputStream;

class Test
{
    public static void main(String
    {
        int a=100, b=0, c;
        c=a/b;
        System.out.println(c);
        System.out.println("hello"
    }
}
```

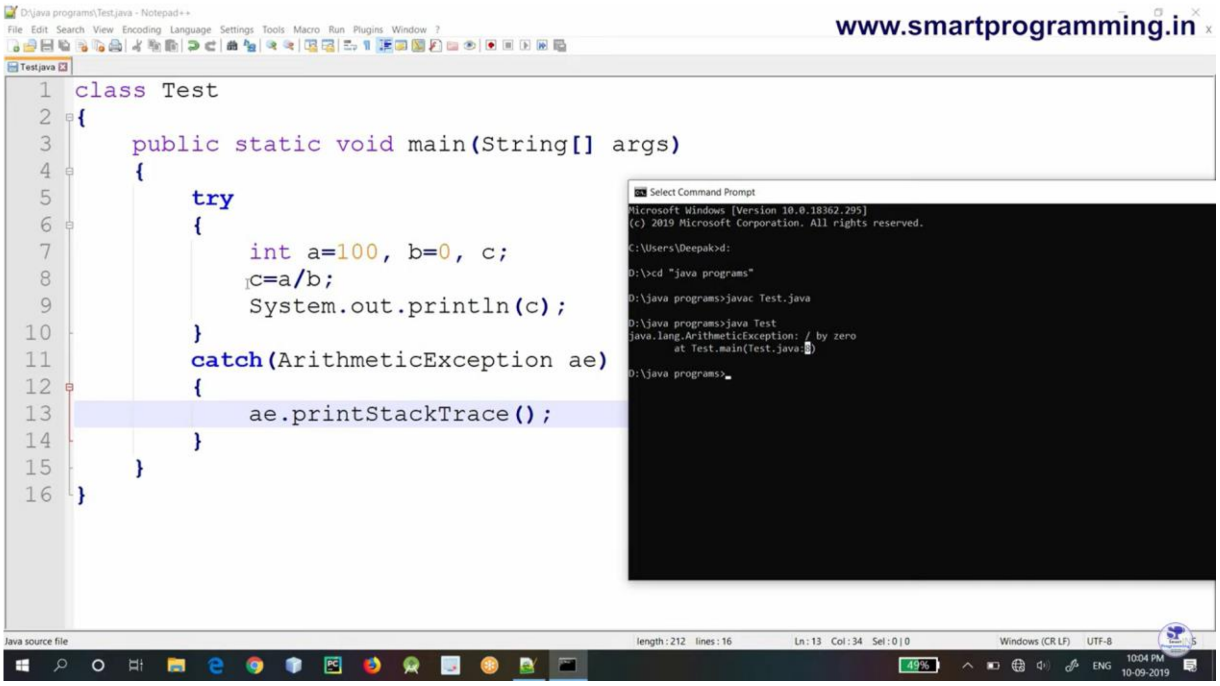
```
D:\java programs>java Test
java.io.FileNotFoundException: d:\abc.txt (The system cannot find the file specified)
hello

D:\java programs>javac Test.java
Test.java:8: error: unreported exception FileNotFoundException; must be caught or declared to be thrown
                FileInputStream fis=new FileInputStream("d:/abc.txt");
                                    ^
1 error

D:\java programs>javac Test.java

D:\java programs>java Test
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
hello

D:\java programs>javac Test.java

D:\java programs>java Test
50
hello

D:\java programs>javac Test.java

D:\java programs>java Test
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Test.main(Test.java:8)

D:\java programs>
```

Refer Examples of Checked n Unchecked Exception.

8) Methods to print Exceptions: -

1) printStackTrace (): -

It prints => Exception Name + Description + Stack Trace



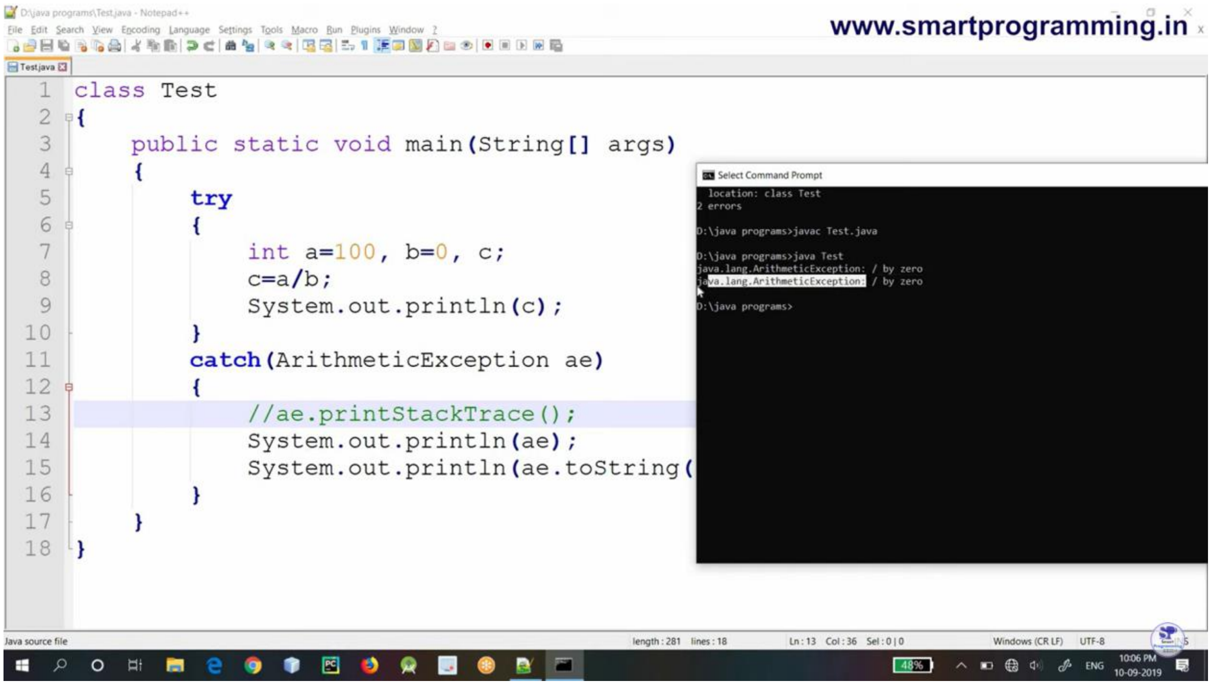
www.smartprogramming.in

```java
class Test
{
    public static void main(String[] args)
    {
        try
        {
            int a=100, b=0, c;
            c=a/b;
            System.out.println(c);
        }
        catch(ArithmeticException ae)
        {
            ae.printStackTrace();
        }
    }
}
```

```
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Deepak>d:

D:\>cd "java programs"

D:\java programs>javac Test.java

D:\java programs>java Test
java.lang.ArithmeticException: / by zero
        at Test.main(Test.java:8)

D:\java programs>_
```

2) System.out.println (): -

It prints=> Exception name + Description



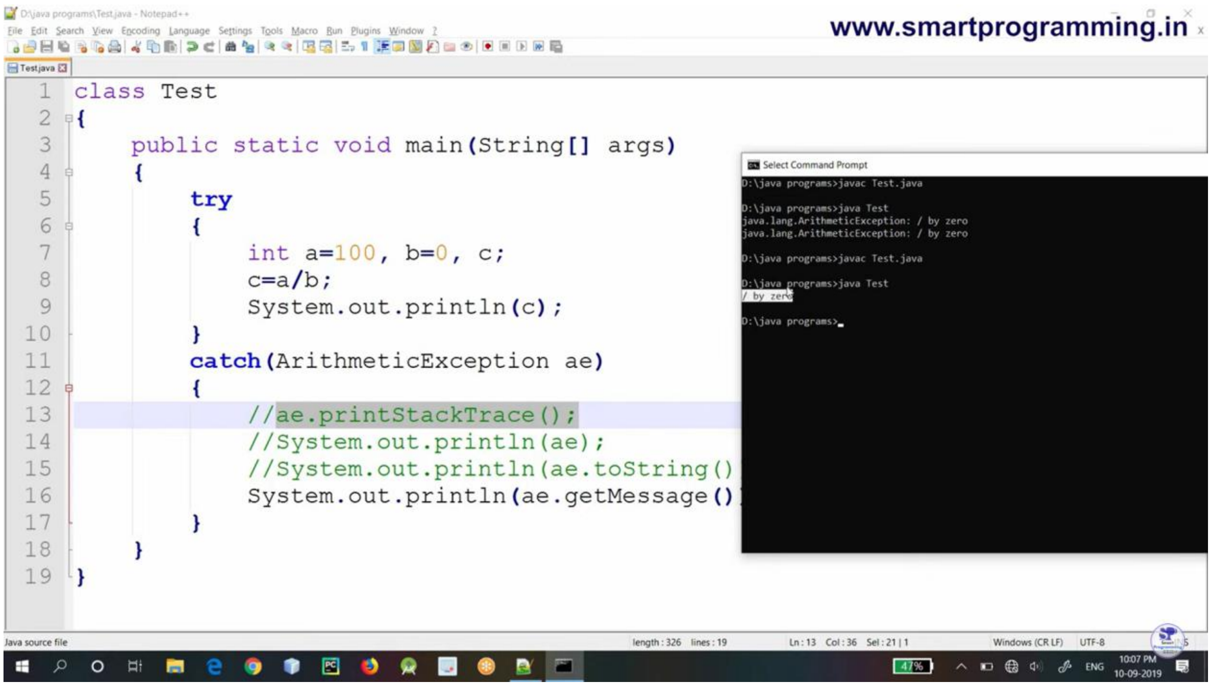
www.smartprogramming.in

```java
class Test
{
    public static void main(String[] args)
    {
        try
        {
            int a=100, b=0, c;
            c=a/b;
            System.out.println(c);
        }
        catch(ArithmeticException ae)
        {
            //ae.printStackTrace();
            System.out.println(ae);
            System.out.println(ae.toString(
        }
    }
}
```

```
  location: class Test
2 errors

D:\java programs>javac Test.java

D:\java programs>java Test
java.lang.ArithmeticException: / by zero
java.lang.ArithmeticException: / by zero

D:\java programs>
```

3) getMessage (): -

It prints=> Description only.



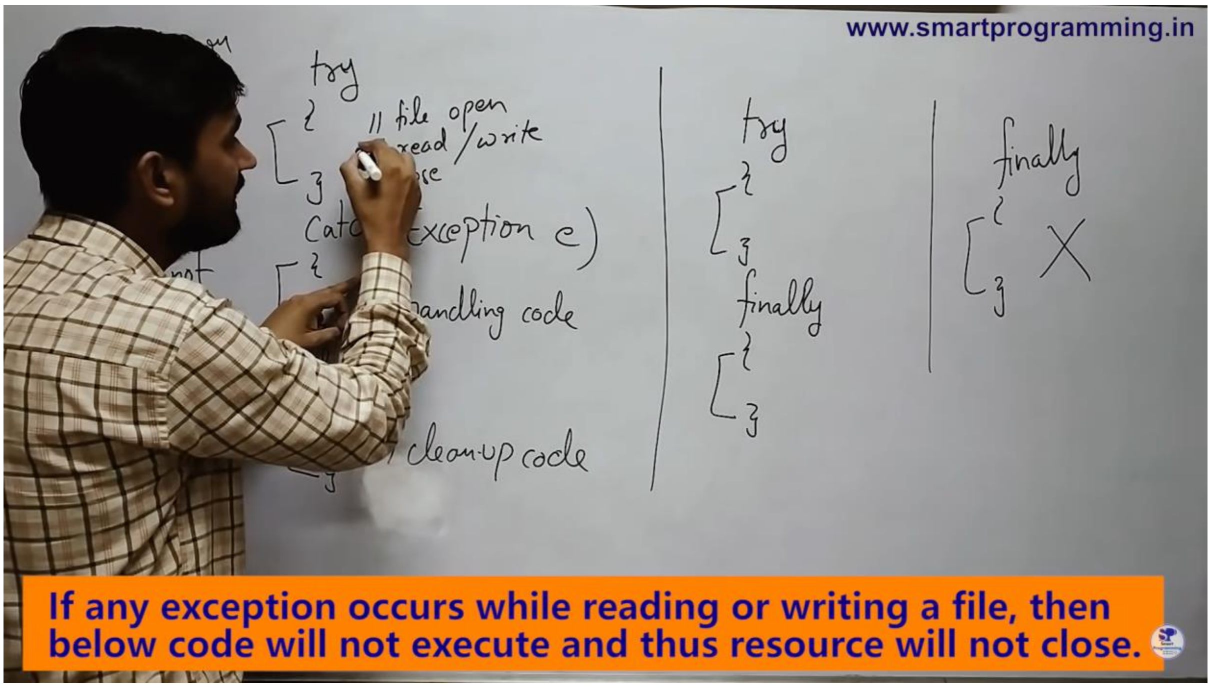
```
class Test
{
    public static void main(String[] args)
    {
        try
        {
            int a=100, b=0, c;
            c=a/b;
            System.out.println(c);
        }
        catch(ArithmeticException ae)
        {
            //ae.printStackTrace();
            //System.out.println(ae);
            //System.out.println(ae.toString()
            System.out.println(ae.getMessage()
        }
    }
}
```

```
D:\java programs>javac Test.java

D:\java programs>java Test
java.lang.ArithmeticException: / by zero
java.lang.ArithmeticException: / by zero

D:\java programs>javac Test.java

D:\java programs>java Test
/ by zero

D:\java programs>_
```

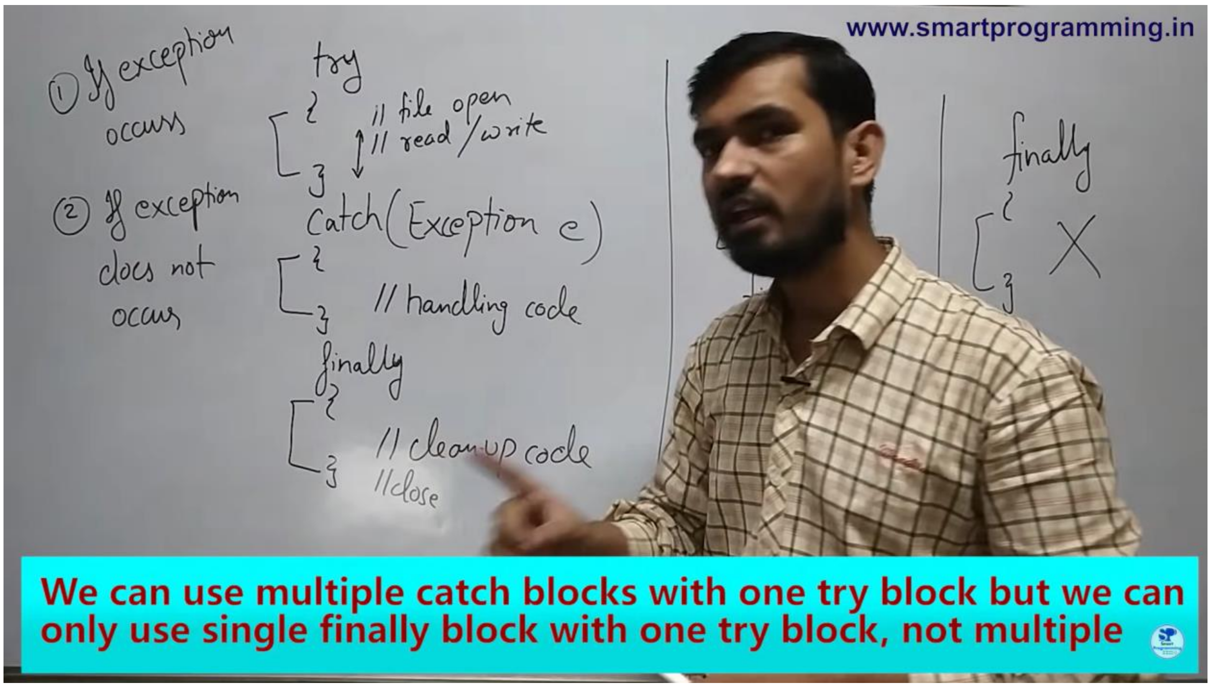
9) Finally Block: -

finally is the block that is always executed whether exception is handled or not

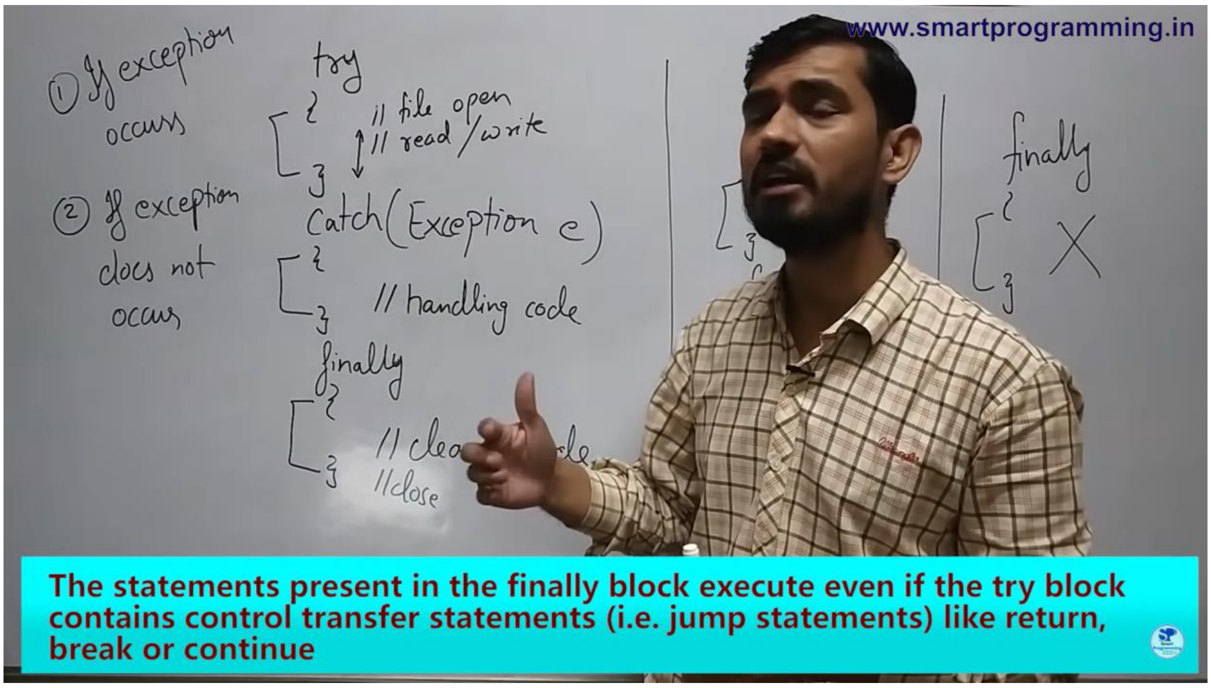
Clean up code/Closing code will be written in finally block



If any exception occurs while reading or writing a file, then below code will not execute and thus resource will not close.



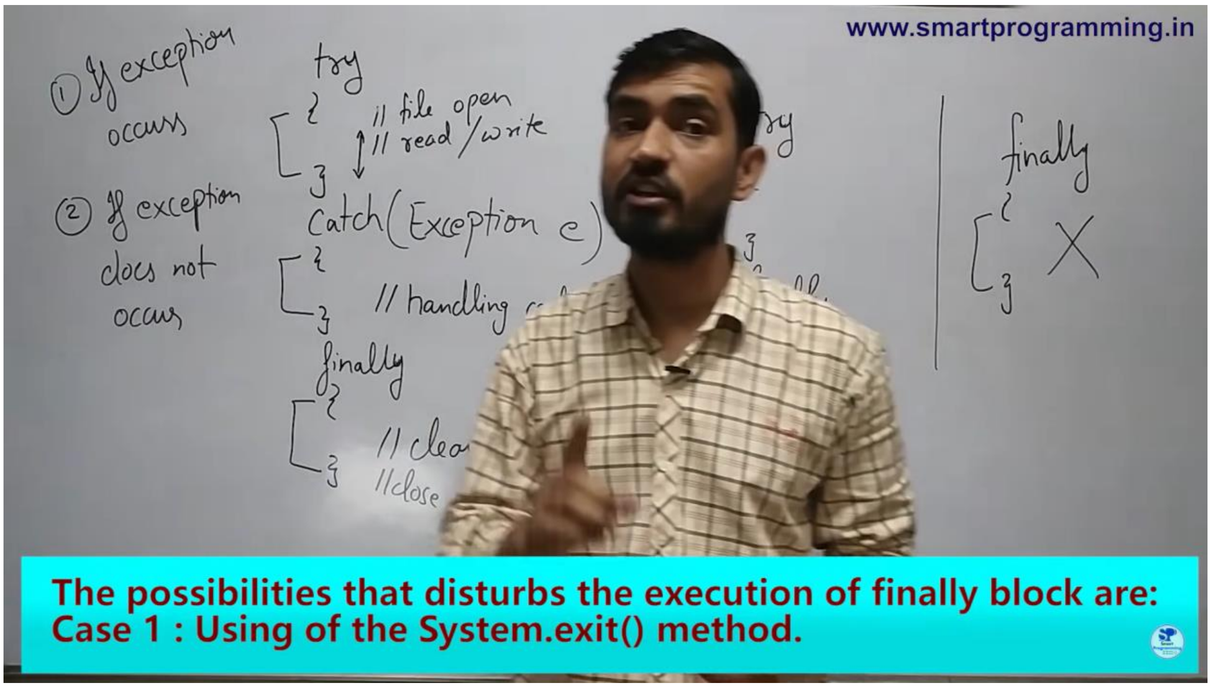
We can use multiple catch blocks with one try block but we can only use single finally block with one try block, not multiple



The statements present in the finally block execute even if the try block contains control transfer statements (i.e. jump statements) like return, break or continue

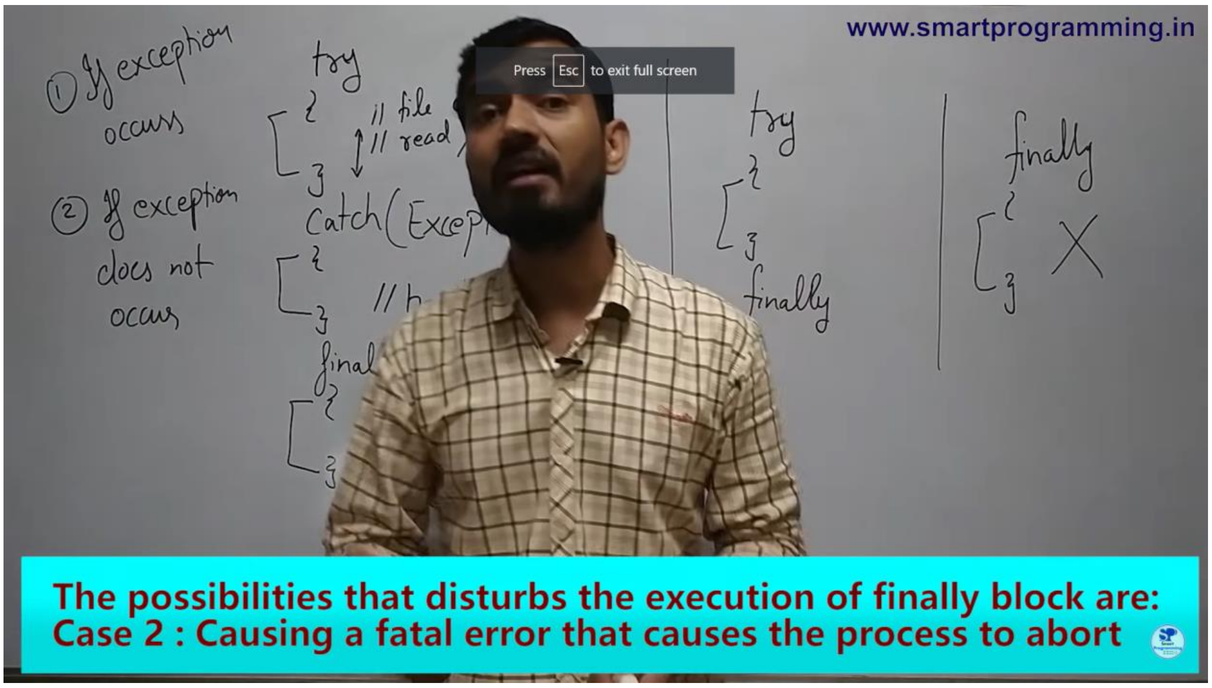
Finally block will not executed if

1)



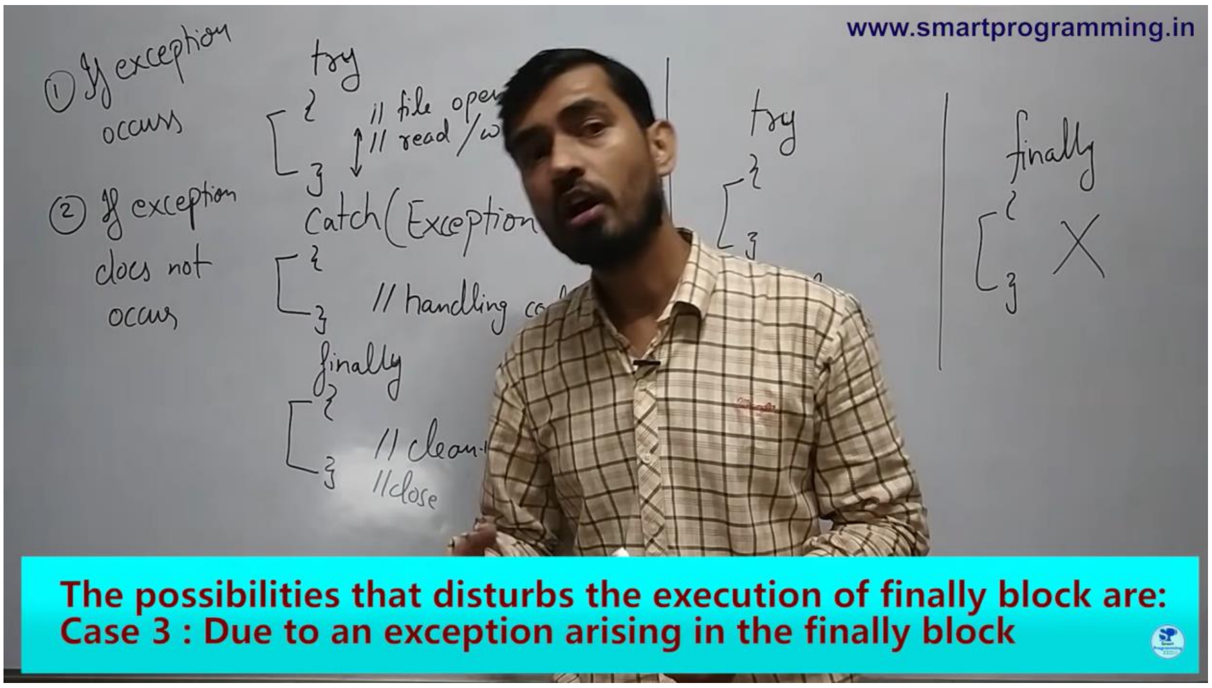
The possibilities that disturbs the execution of finally block are:
Case 1 : Using of the System.exit() method.

2)


www.smartprogramming.in
Press Esc to exit full screen
① If exception occurs
② If exception does not occur
try { // file // read } catch(Excep { //h } final { }
try { } finally
finally { } X

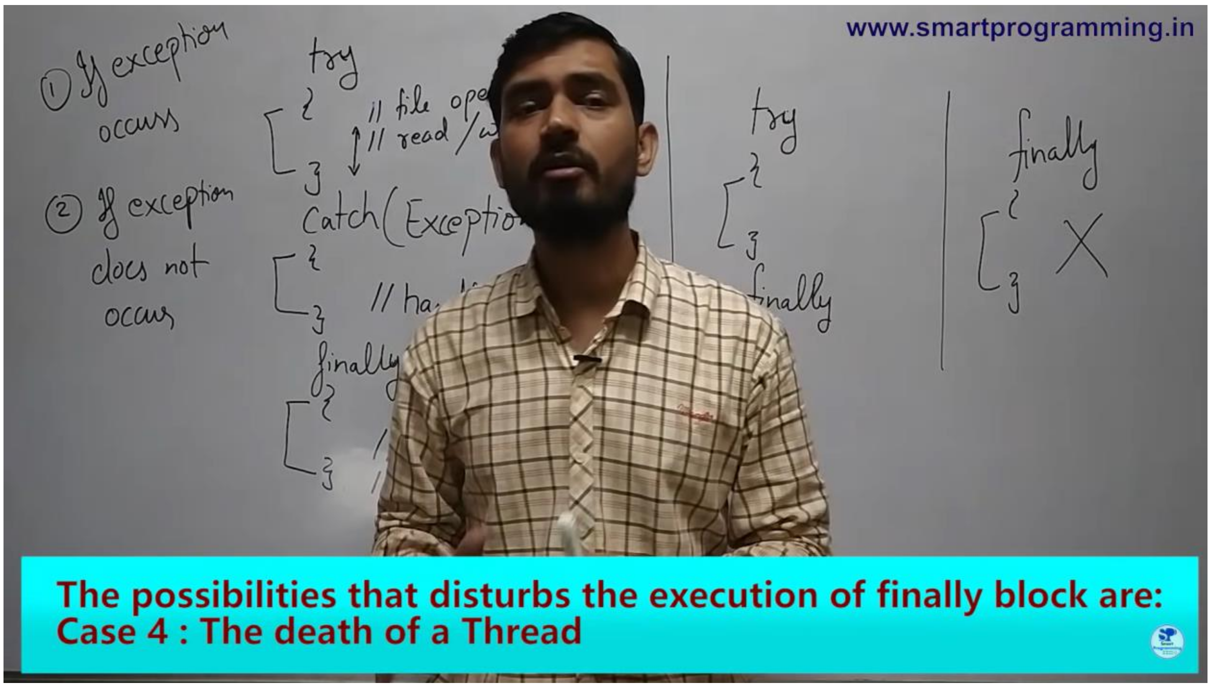
The possibilities that disturbs the execution of finally block are:
Case 2 : Causing a fatal error that causes the process to abort

3)


www.smartprogramming.in
① If exception occurs
② If exception does not occur
try { // file open // read / w } catch(Exception { // handling co } finally { // clean // close }
try { }
finally { } X

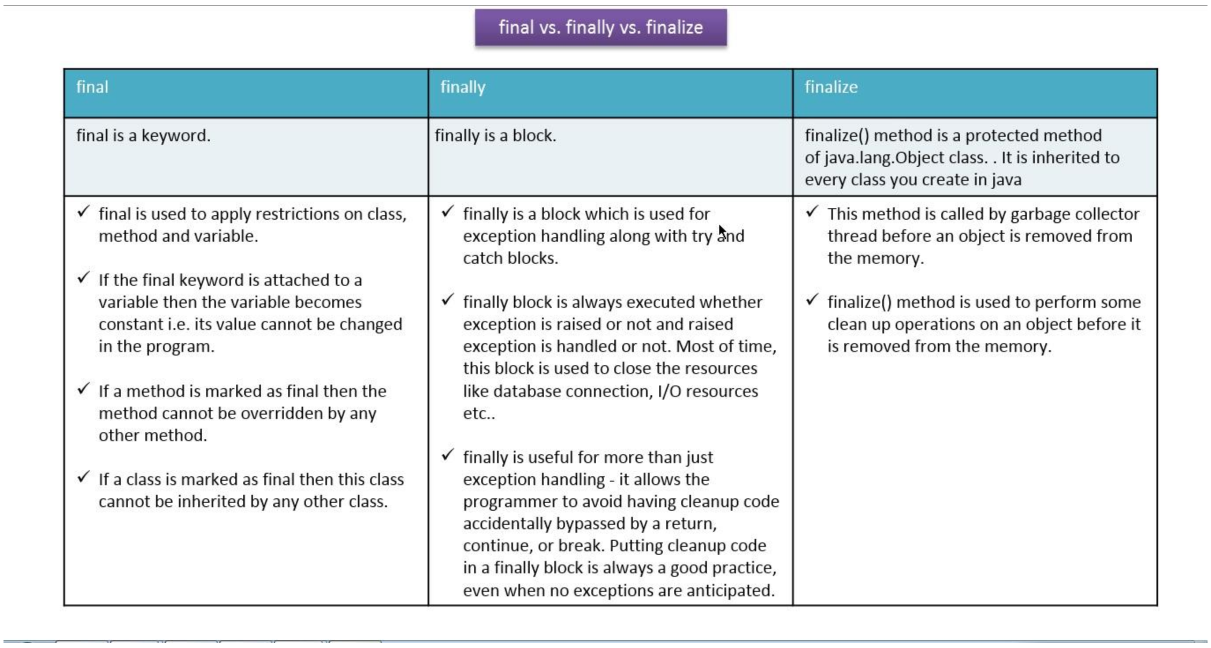
The possibilities that disturbs the execution of finally block are:
Case 3 : Due to an exception arising in the finally block

4)



The possibilities that disturbs the execution of finally block are:
Case 4 : The death of a Thread

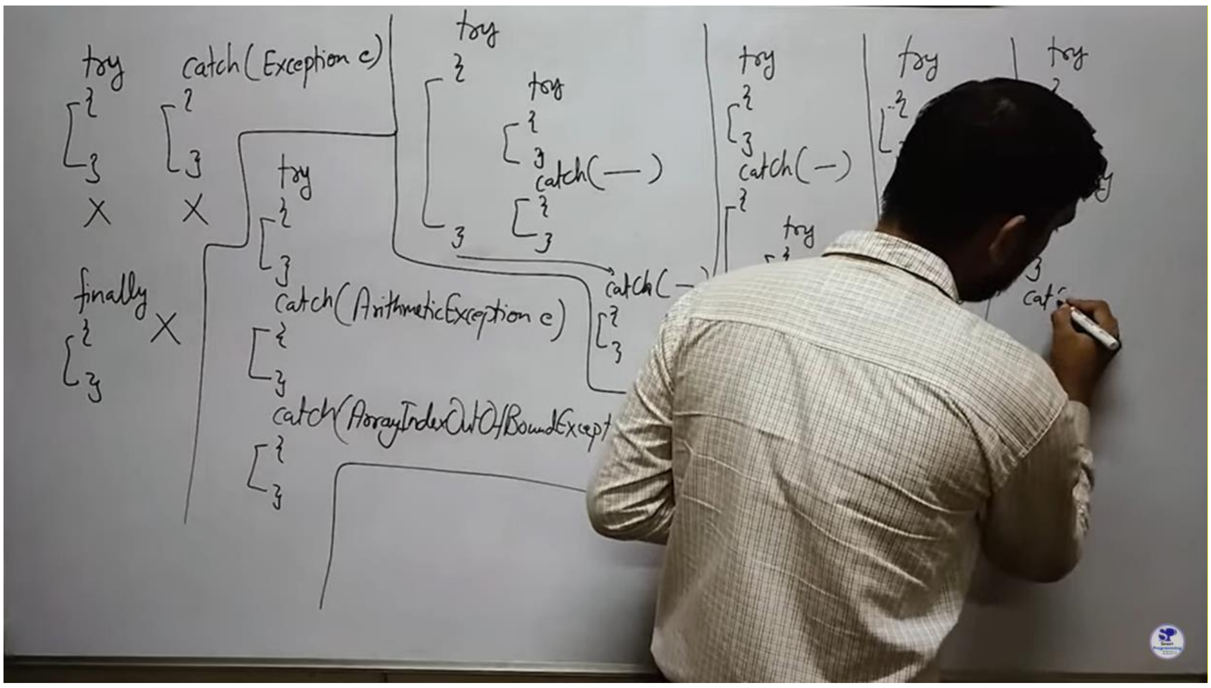
Difference between final finally & finalize!

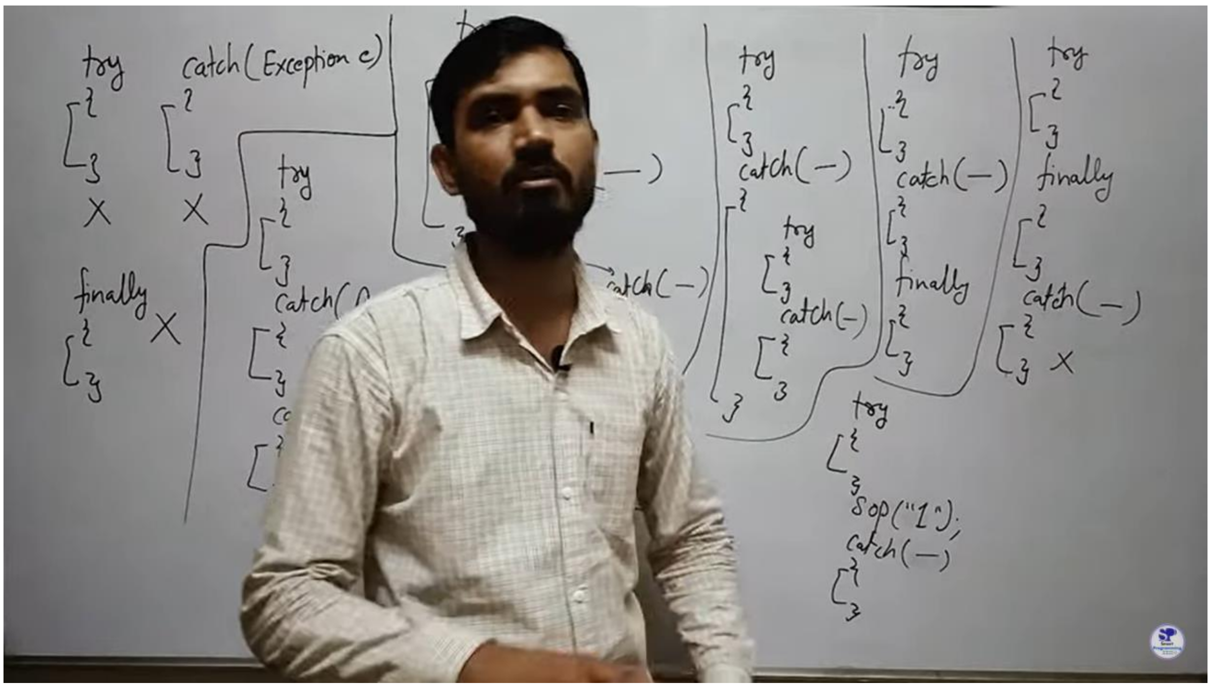


final vs. finally vs. finalize

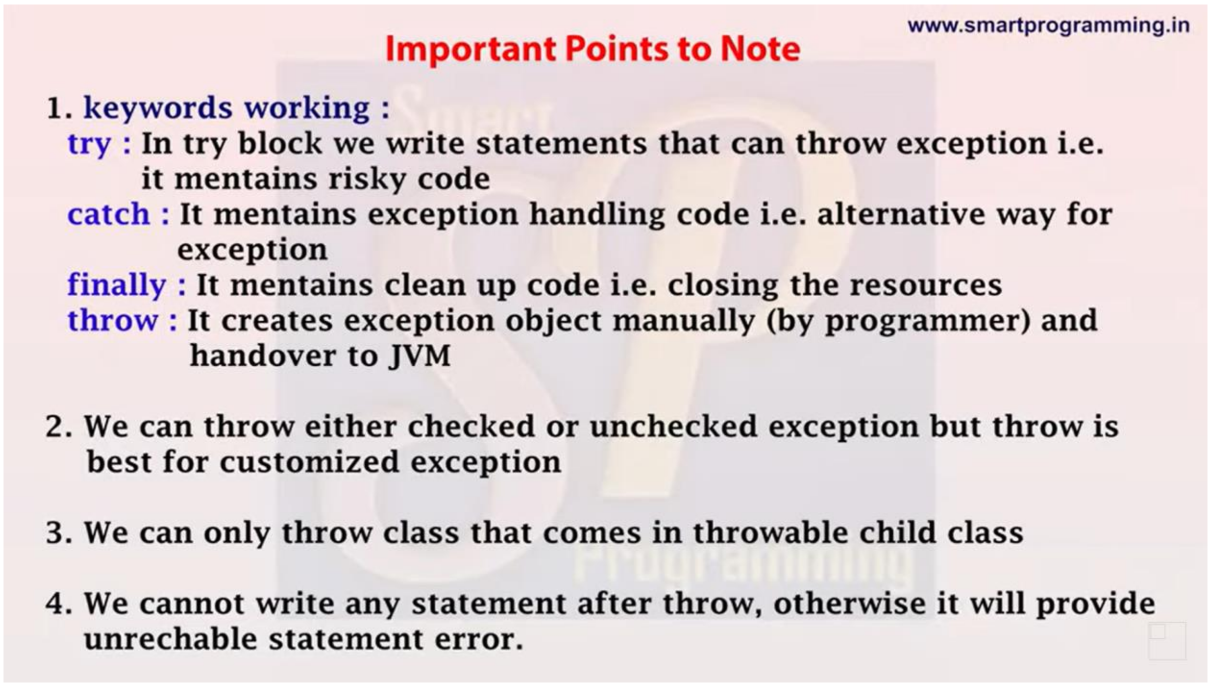
final	finally	finalize
final is a keyword.	finally is a block.	finalize() method is a protected method of java.lang.Object class. . It is inherited to every class you create in java
✓ final is used to apply restrictions on class, method and variable. ✓ If the final keyword is attached to a variable then the variable becomes constant i.e. its value cannot be changed in the program. ✓ If a method is marked as final then the method cannot be overridden by any other method. ✓ If a class is marked as final then this class cannot be inherited by any other class.	✓ finally is a block which is used for exception handling along with try and catch blocks. ✓ finally block is always executed whether exception is raised or not and raised exception is handled or not. Most of time, this block is used to close the resources like database connection, I/O resources etc.. ✓ finally is useful for more than just exception handling - it allows the programmer to avoid having cleanup code accidentally bypassed by a return, continue, or break. Putting cleanup code in a finally block is always a good practice, even when no exceptions are anticipated.	✓ This method is called by garbage collector thread before an object is removed from the memory. ✓ finalize() method is used to perform some clean up operations on an object before it is removed from the memory.

Various Possible try Catch finally Combinations :-


try { } ×
catch(Exception e) { } ×
finally { } ×
try { } catch(ArithmeticException e) { } catch(ArrayIndexOutOfBoundExcept
try { try { } catch(—) { } } catch(—) { }
try { } catch(—) { try


try { } ×
catch(Exception e) { } ×
finally { } ×
try { } catch(—) { try { } catch(—) { } }
try { } catch(—) { } finally { }
try { } finally { } catch(—) { } ×
try { } Sop("1"); catch(—) { }

Throw Keyword in Java Exception Handling



www.smartprogramming.in

Important Points to Note

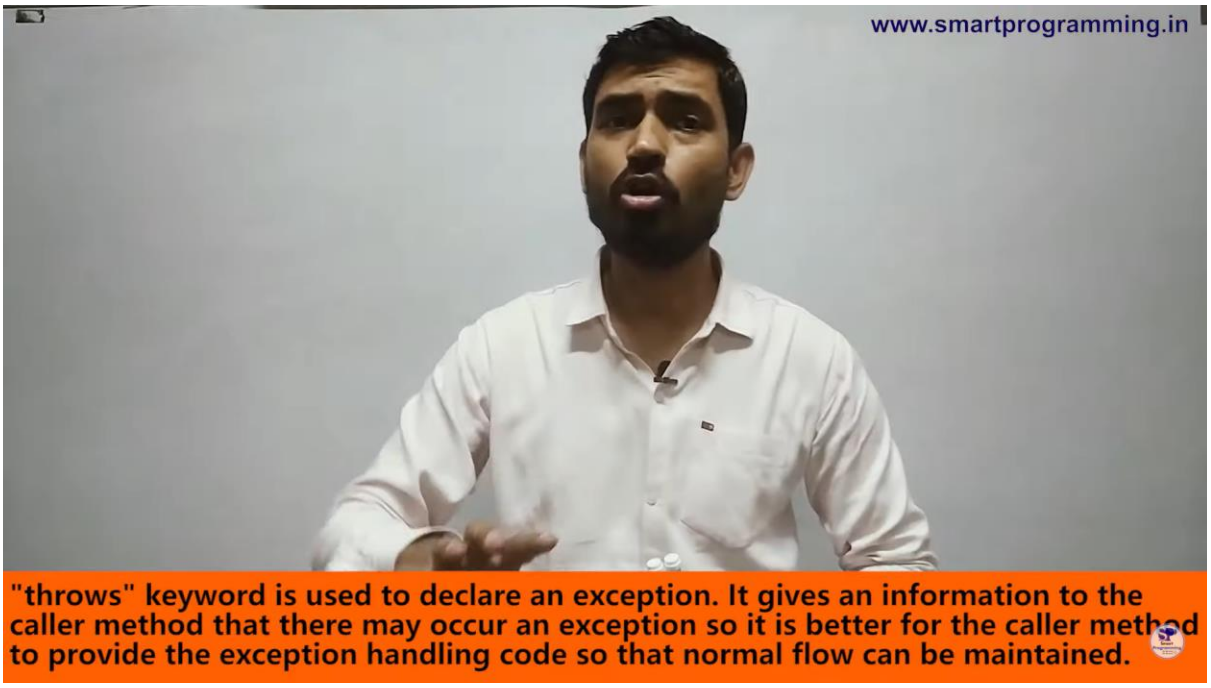
1. keywords working :
   - try : In try block we write statements that can throw exception i.e. it mentains risky code
   - catch : It mentains exception handling code i.e. alternative way for exception
   - finally : It mentains clean up code i.e. closing the resources
   - throw : It creates exception object manually (by programmer) and handover to JVM

2. We can throw either checked or unchecked exception but throw is best for customized exception

3. We can only throw class that comes in throwable child class

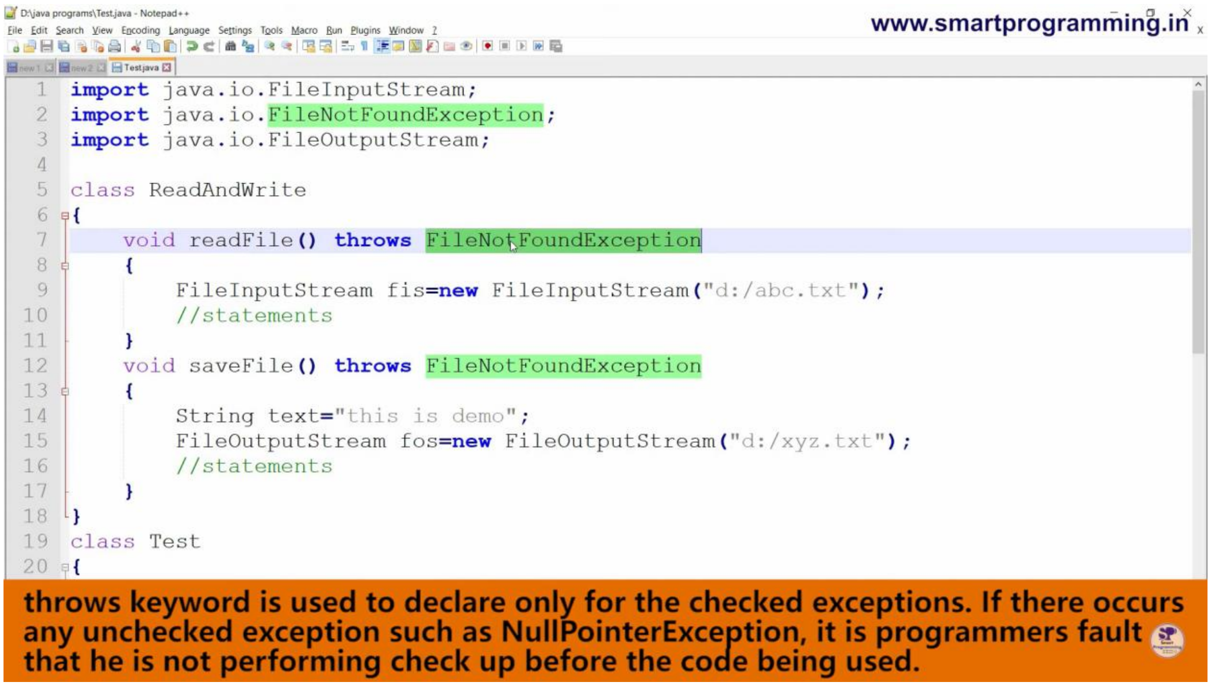
4. We cannot write any statement after throw, otherwise it will provide unrechable statement error.

Throws: -



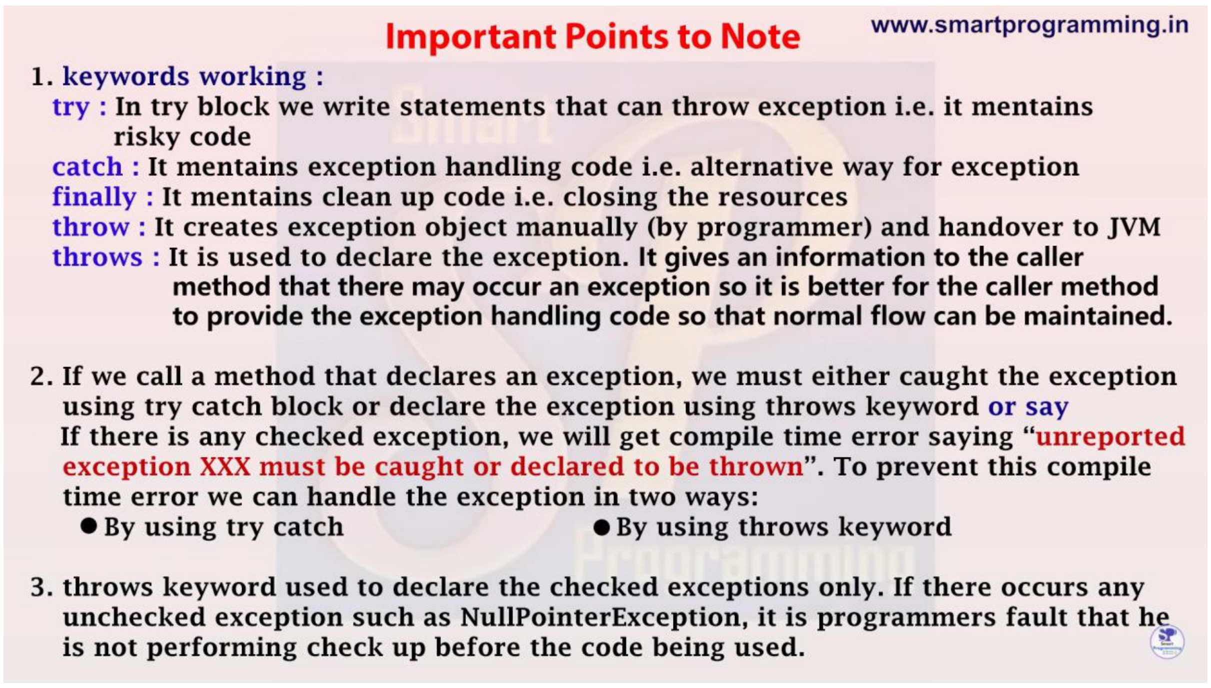
www.smartprogramming.in

"throws" keyword is used to declare an exception. It gives an information to the caller method that there may occur an exception so it is better for the caller method to provide the exception handling code so that normal flow can be maintained.


D:\java programs\Test.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
www.smartprogramming.in

```java
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;

class ReadAndWrite
{
    void readFile() throws FileNotFoundException
    {
        FileInputStream fis=new FileInputStream("d:/abc.txt");
        //statements
    }
    void saveFile() throws FileNotFoundException
    {
        String text="this is demo";
        FileOutputStream fos=new FileOutputStream("d:/xyz.txt");
        //statements
    }
}
class Test
{
```

throws keyword is used to declare only for the checked exceptions. If there occurs any unchecked exception such as NullPointerException, it is programmers fault that he is not performing check up before the code being used.


www.smartprogramming.in

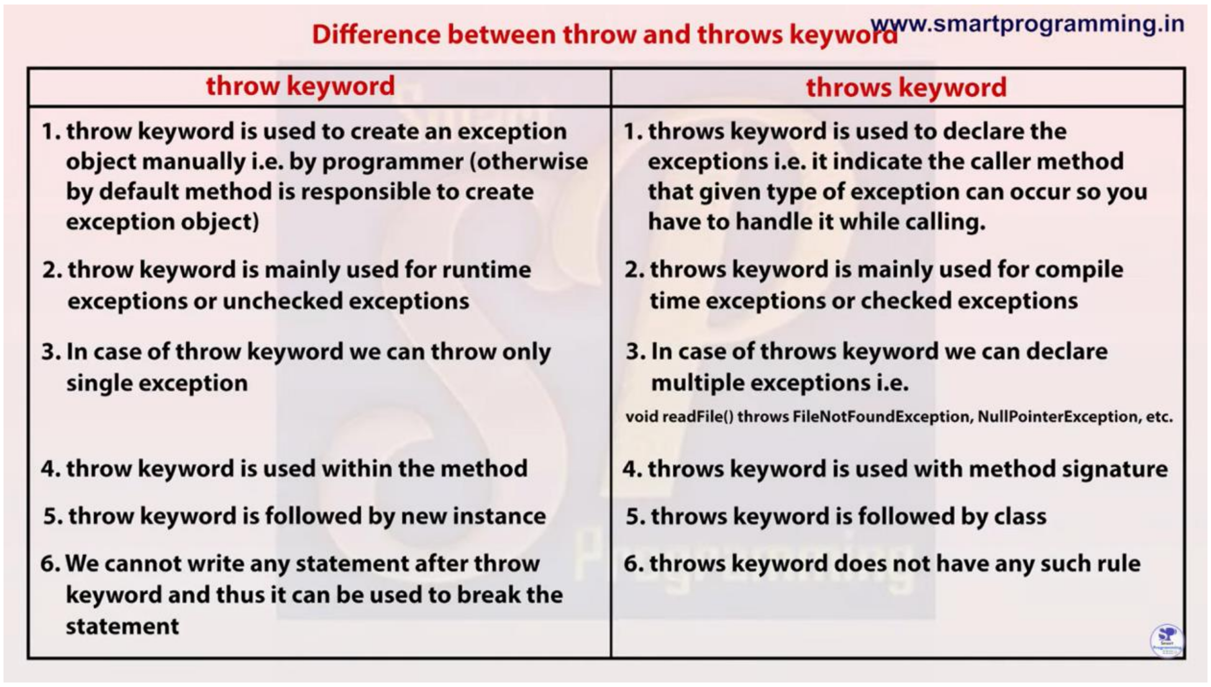
Important Points to Note

1. **keywords working :**
   - **try :** In try block we write statements that can throw exception i.e. it mentains risky code
   - **catch :** It mentains exception handling code i.e. alternative way for exception
   - **finally :** It mentains clean up code i.e. closing the resources
   - **throw :** It creates exception object manually (by programmer) and handover to JVM
   - **throws :** It is used to declare the exception. It gives an information to the caller method that there may occur an exception so it is better for the caller method to provide the exception handling code so that normal flow can be maintained.

2. If we call a method that declares an exception, we must either caught the exception using try catch block or declare the exception using throws keyword or say
   If there is any checked exception, we will get compile time error saying "unreported exception XXX must be caught or declared to be thrown". To prevent this compile time error we can handle the exception in two ways:
   - By using try catch
   - By using throws keyword

3. throws keyword used to declare the checked exceptions only. If there occurs any unchecked exception such as NullPointerException, it is programmers fault that he is not performing check up before the code being used.

Diff between throw and throws keywords



Difference between throw and throws keyword www.smartprogramming.in

throw keyword	throws keyword
1. throw keyword is used to create an exception object manually i.e. by programmer (otherwise by default method is responsible to create exception object)	1. throws keyword is used to declare the exceptions i.e. it indicate the caller method that given type of exception can occur so you have to handle it while calling.
2. throw keyword is mainly used for runtime exceptions or unchecked exceptions	2. throws keyword is mainly used for compile time exceptions or checked exceptions
3. In case of throw keyword we can throw only single exception	3. In case of throws keyword we can declare multiple exceptions i.e. void readFile() throws FileNotFoundException, NullPointerException, etc.
4. throw keyword is used within the method	4. throws keyword is used with method signature
5. throw keyword is followed by new instance	5. throws keyword is followed by class
6. We cannot write any statement after throw keyword and thus it can be used to break the statement	6. throws keyword does not have any such rule

-------------xxxx---------xxxx---------xxxx------------