**Throw:**

....

static{

try

{

throw new Exception("Something went wrong!!");

}

catch (Exception exp) {

System.out.println("Error: "+exp.getMessage());

}

}

....

**Throws:**

public void sample() throws ArithmeticException{

//Statements

.....

//if (Condition : There is an error)

ArithmeticException exp = new ArithmeticException();

throw exp;

...

}

By using **Throw keyword** in java you cannot throw more than one exception but using **throws** you can declare multiple exceptions. PFB the examples.

for e.g.

**Throw:**

throw new ArithmeticException("An integer should not be divided by zero!!")

throw new IOException("Connection failed!!")

**Throws:**

throws IOException, ArithmeticException, NullPointerException,

ArrayIndexOutOfBoundsException

Imagine you have been assigned a task of finding a specific book, and then reading and explaining its contents to a class of students. The required sequence may look like:

* Get the specified book
* Read aloud its contents
* Explain the contents to a class of students.

But what happens if you can't find the specified book? You can't proceed with the rest of the action without it so you need to report back to the person who assigned the task to you. This unexpected event (missing book) prevents you from completing your task. By reporting it back, you want the originator of this request to take corrective or alternate steps.

import java.io.FileNotFoundException;  
class DemoThrowsException {  
    public void readFile(String file) throws FileNotFoundException

{            boolean found = findFile(file);                           
        if (!found)  
            throw new FileNotFoundException("Missing file");      
        else {  
            //code to read file  
        }  
    }  
    boolean findFile(String file) {  
        //code to return true if file can be located  
    }  
}

|  |  |  |
| --- | --- | --- |
| **No.** | **throw** | **throws** |
| 1) | Java throw keyword is used to explicitly throw an exception. | Java throws keyword is used to declare an exception. |
| 2) | Checked exception cannot be propagated using throw only. | Checked exception can be propagated with throws. |
| 3) | Throw is followed by an instance. | Throws is followed by class. |
| 4) | Throw is used within the method. | Throws is used with the method signature. |
| 5) | You cannot throw multiple exceptions. | You can declare multiple exceptions e.g. public void method()throws IOException,SQLException. |