Difference between throw and throws in Java

Without wasting any more of your time, here is a list of key differences between the throw and throws keywords in Java. Even though both are related to errors and exceptions in Java, there are some subtle differences that are listed here.

1. Purpose

The `throw` keyword is used to throw Exception from any method or static block in Java while throws keyword, used in the method declaration, denoted which Exception can possibly be thrown by this method.

2.Handling

If any method throws a checked Exception as shown in the below Example, then the caller can either handle this exception by catching it or can re-throw it by declaring another throws clause in the method declaration.

```
public void read() throws IOException{
   throw new IOException();
}
```

failure to either catch or declaring throws in method signature will result in compile time error.

3. Usage

The `throw` keyword can be used in the switch case in Java but the throws keyword can not be used anywhere except on the method declaration line.

4. Overloading

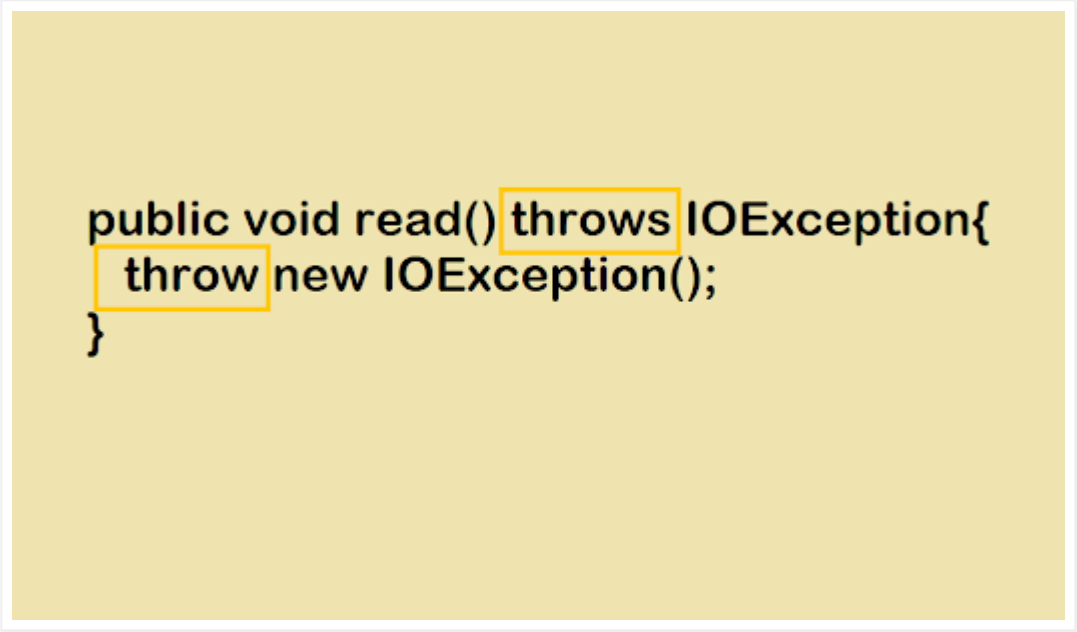
As per Rules of overriding in Java, the overriding method can not throw Checked Exception higher in the hierarchy than the overridden method. This is the rule for throws clause while overriding a method in Java.

5. Eligibility

Both Checked and Unchecked Exception can be declared to be thrown using the throws clause in Java.

6. Control

The throw keyword transfers control to the caller while throws are suggesting for information and compiler checking. See these free Java Programming courses to learn more about throw, throws, and in-general exception handling in Java.



That's all on the **difference between throw vs throws in Java** and Exception handling. You must try some examples to use throw and throws as well and rather importantly you must know when to use the throw and throws keyword in Java. In summary, use throw to actually throw an exception which will give control back to the caller, and use throws to declare which Exception can be thrown by a particular method, which allows the caller to handle them.