Propagation of a thrown object

Consider a program that has a number of methods. Suppose the program is executed by calling method main. Method main ends up calling method first, and then first calls method second.

```
public static void main(String[] args) {
    try {
        …
        first();
    }
    catch (…) { … }
}
public static void first() {
    second();
}
public static void second() { … }
```

Now suppose that an object a0 is thrown, signaling some sort of error, and suppose that no try-statements are being executed. Since there is no catch clause to catch the object, it is thrown further, to the calling method. Thus, in this example, it appears that the call on method second throws a0. And if this call on method second does not appear within a try-block, thrown out further, to calling method main. So it looks like the method call throws a0.

This call on first is within a try-block, so if one of its catch-clauses catches a0, that's the end of the throwing.

But suppose that the catch-blocks of the try-statement do not catch the object. Then it is thrown outside the try-statement, so that it looks like the try-statement throws the object. In our example, the try-statement is not in a try-block, so rule 1 is used: object a0 is thrown to the caller of method main, which is within the Java system itself.

The call of method main within the system is within the try-block of a try-statement that catches all Throwable objects, so the catch-block is guaranteed to print the information about the thrown object.

Here's a summary of the rules, assuming that the statement in question is in method m called by method c:

- If a statement not in a try-block throws an object, it is thrown to method c. The effect is that the call of m in c threw it.
- If a statement in a try-block throws an object an a catch clause catches it, the corresponding catch-block is executed. If it terminates normally, execution of the try-statement terminates as well. If it throws an exception, then the rules here determine where it is thrown.
- If a statement in a try-block throws an object but a catch clause does not catch it, the effect is that the try-statement itself throws it further.