

**NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.**

**SCENARIO:**

See the [J02-File.renameTo](#) puzzle.

```
01  // OMITTED: Import whatever is needed
02  public final class FileUtils {
03      public static boolean move (File source, File destination,
04          boolean overwrite) {
05          if (overwrite) {
06              if (destination.exists())
07                  destination.delete();
08              return source.renameTo(destination);
09          }
10          else if (!destination.exists()) {
11              return source.renameTo(destination);
12          }
13          else {
14              return false;
15          }
16      }
17  }
```

Questions:

See the [J02-File.renameTo](#) puzzle.

**NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.**

**ANSWER:**

a

See the [J02-File.renameTo](#) puzzle.

**NOTE: THE REST OF THIS DOCUMENT CONTAINS EXTRA INFORMATION FOR THE PROJECT RESEARCHERS. IT IS NOT TO BE SHOWN TO THE PUZZLE TAKERS.**

**TAGS:**

java, file-class, io-operation, file-operation, ignoring-return-value, deprecated-api

**CATEGORIES:**

Blindspot - NO

Type - File

Number of distinct functions - 6

Number of total functions - 9

Blindspot function - NA

Function call omitted - NA

Blindspot type - NA

Number of Parameters in the blindspot function - NA

Cyclomatic complexity - 6

**NAME:**

File class, renameTo method - An abstract representation of file and directory pathnames.

**MORE INFORMATION:**

To see an incorrect way to use the API, look at the [J02-File.renameTo](#) puzzle.

**REFERENCES:**

See the [J02-File.renameTo](#) puzzle.