ToDo: mark each item with

- X if condition is valid
- NA if condition is not applicable
- U if conditions is not fulfilled

1. Specification / Design	
a. Is the functionality described in the specification fully implemented by the code?	U
b. Is there any excess functionality in the code but not described in the specification?	X
2. Initialization and Declarations	
a. Are all local and global variables initialized before use?	NA
b. Are variables and class members of the correct type and are the using appropriate modifiers?	U
c. Are variables declared in the proper scope?	X
d. Is a constructor called when a new object is desired?	X
e. Are Variable names spelled correctly and consistently	U
f. Make sure that primitive data types are not set to null or empty	X
g. Is 'static' keyword used correctly?	NA
h. Is every variable correctly typed?	X
i. Are there variables that should be constants?	X
3. Method Definition	
a. Are descriptive method names used in accord with the naming convention?	U
b. Is every method parameter value checked before being used?	U
c. Are there static methods that should be non-static or vice-versa?	U

4. Class Definition	
a. Does each class have an appropriate constructor?	U
5. Date References	
a. For every object or array reference: is the value certain to be non-null?	X
6. Method Calls	
a. Are parameters presented in the correct order?	X
b. Are parameters of the proper type for the method being called?	X
c. Are method return values used properly? Cast to the required type?	X
d. When calling a method that has a return value, be sure to use the return value properly.	X
7. Arrays	
a. Are there any off-by-one errors in array indexing?	X
b. Can array indexes ever go out-of-bounds?	X
c. Are array declarations syntactically correct?	X
d. Are the row and column being indexed in the right order for a 2D array?	X
8. Object Comparison	
a. Are all objects (including Strings) compared with "equals" and not "=="?	X
9. Output Format	
a. Are there any spelling or grammatical errors in displayed output?	NA
b. Is the output formatted correctly in terms of line stepping and spacing?	X

10. Computation, Comparison and Assignment	
a. Check order of computation/evaluation, operator precedence and parenthesizing	X
b. Can the denominator of a division ever be zero?	NA
c. Is integer arithmetic, especially division, ever used inappropriately, causing unexpected truncation/rounding?	NA
d. Check each condition to be sure the proper relational and logical operators are used.	X
e. If the test is an error-check, can the error condition actually be legitimate in some cases?	NA
f. Does the code rely on any implicit type conversions?	X
g. Are there any computations with mixed data types?	X
h. Is overflow/underflow possible during computation?	NA
i. For each expression with more than one operator: are the assumptions about order of evaluation and precedence correct?	NA
j. Are there any comparisons between variables of inconsistent types?	X
k. Are the comparison operators correct?	X
11. Exceptions	
a. Are all relevant exceptions caught?	NA
b. Is the appropriate action taken for each catch block?	NA
12. Flow of Control	
a. Check that nested if statements don't have “dangling else” problems.	X
b. Are all loops correctly formed, with the appropriate initialization, increment and termination expressions?	X
c. Are open-close parentheses and brace pairs properly situated and matched?	X
d. Do logical expressions evaluate to the correct true or false value?	X

e. Do Boolean functions return the correct value?	X
f. Has each boolean expression been simplified by driving negations inward?	X
g. For every boolean test: Is the correct condition checked?	X
h. Is each Boolean expression correct?	X
i. Are there improper and unnoticed side-effects of a comparison?	X
j. Has an "&" inadvertently been interchanged with a "&&" or a "|" for a "||"?	X
k. Does the code avoid comparing floating-point numbers for equality (IEEE 754)?	NA
l. Is every three-way branch (less,equal,greater) covered?	NA
m. For each loop: Is the best choice of looping constructs used?	J
n. Will all loops terminate?	X
o. When there are multiple exits from a loop: is each exit necessary and handled properly?	X
p. In a switch statement is every case terminated by break or return?	NA
q. Do all switch statements have a default branch?	NA
r. Are missing switch case break statements correct and marked with a comment?	NA
s. Is the nesting of loops and branches too deep, and is it correct?	X
t. Can any nested if statements be converted into a switch statement?	X
u. Are null bodied control structures correct and marked with braces or comments?	X
v. Does every method always terminate?	X
w. Do named break statements send control to the right place	NA

1.a. (1.1) Keine K. für die Darstellung der Wahrheitstabellen
Kein Generator (keine Factory)
Keine Zugriffsmethoden auf die Bestandteile
Die gelbe Tabelle fehlt (im Test).
1.3.a. Tabelle aus den Folien fehlt.

1.b. kein excess functionality
2.a. keine globale Variablen
2.b. alles ist public
2.e. mmbve
getmmve
2.g. no static
3.a. getmmve
3.b. wird nicht gechecket
3.c. es hätte alles static sein können
4.a. kein Constructor
9.a. kein geschriebener Output
10.b. keine Division
10.c. keine Integer-Arithmetik

10.e. keine Error Checks

10.h. nicht testbar

10.i. keine Ausdrücke mit mehreren Operatoren

11.a., b. keine Exceptions

12.k. kein floating point comparison

12.l. keine 3way branches

12.m. in printTable hätte foreach genutzt werden können

12.p.19.ii. keine switch statements

12.w. keine named break statements