

## Übung 005

### Aufgabe 1: De Morganschen Gesetze

Die **De Morganschen Gesetze** lauten

$$\neg(A \wedge B) = \neg A \vee \neg B$$

$$\neg(A \vee B) = \neg A \wedge \neg B$$

bzw. in Prosa

$$\textit{nicht} (A \textit{ und } B) = (\textit{nicht } A) \textit{ oder } (\textit{nicht } B)$$

$$\textit{nicht} (A \textit{ oder } B) = (\textit{nicht } A) \textit{ und } (\textit{nicht } B)$$

Beweise die **De Morganschen Gesetze** durch das Ausfüllen der folgenden Tabellen. Nutze dazu die in im YouTube vorgestellten Operatoren und ermittle die entsprechenden Aussagen (true, false) durch Ausgabe des Ergebnisses.

A	B	$\neg(A \wedge B)$	$\neg A \vee \neg B$
true	true		
true	false		
false	true		
false	false		

A	B	$\neg(A \vee B)$	$\neg A \wedge \neg B$
true	true		
true	false		
false	true		
false	false		

## Aufgabe 2: Gleichheitsoperator

Wie im YouTube Video angesprochen ist der Gleichheitsoperator bei Gleitkommatypen keine sichere Variante, um die Gleichheit zweier Variablen zu ermitteln. Entwickeln sie mit Hilfe der vorgestellten *numeric\_limits* Bibliothek ein Verfahren um Gleitkommatypen vergleichen zu können. Sie können dazu mit dem folgenden Programmcode starten.

---

```
1  #include <cmath>
2  #include <iomanip>
3  #include <iostream>
4  #include <limits>
5
6  int main()
7  {
8
9      double x1 = 3.0;
10     double x2 = std::sqrt(3.0) * std::sqrt(3.0);
11
12     bool equal_check = (x1 == x2);
13
14     std::cout << "x1: " << std::setprecision(20) << x1 << std::endl;
15     std::cout << "x2: " << std::setprecision(20) << x2 << std::endl;
16     std::cout << "equal check: " << equal_check << std::endl;
17
18     std::cout << "epsilon: " << std::numeric_limits<double>::epsilon()
19               << std::endl;
20
21     return 0;
22 }
```

---