



Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 1 – 22.04.2024

Abgabe: Bis 28.04.2024, 23:59 Uhr per E-Mail an Ihre/n Tutor/in

Homepage zur Vorlesung:

<https://aam.uni-freiburg.de/agdo/lehre/ss24/prog/index.html>

Aufgabe 1 (5 Punkte). In C++ ist es wichtig darauf zu achten, in welcher Reihenfolge verschiedene Operationen ausgewertet werden und welchen Typ Variablen haben. Es seien folgende Variablen definiert:

```
int i=6; double x=4.5; int j=x; bool A=false; bool B=true;
```

Bestimmen Sie den Typ (`int`, `double`, `bool`) jedes der folgenden Ausdrücke und bestimmen Sie den Zahlen- bzw. Wahrheitswert des Ausdrucks.¹

(i) $i/3$	(ii) $2*i/2$	(iii) $i++/2.0$	(iv) $++i/2$
(v) $A \ \&\& \ !A$	(vi) $A \ \&\& \ B$	(vii) $!!A$	(viii) $(i < x) \ \&\& \ A$
(ix) j	(x) $++i/4*j$		

Vor jedem Ausdruck sollen die Werte sein wie in der Aufgabenstellung.

Aufgabe 2 (5 Punkte). Schreiben Sie ein Programm, welches zwei ganze Zahlen in der Konsole einliest und diese in zwei verschiedene Variablen speichert. Das Programm soll dann die Werte der beiden Variablen vertauschen und die getauschten Werte in der Konsole ausgeben.

Aufgabe 3 (5 Punkte). Eine Zahl heißt *perfekt* wenn sie die Summe ihrer Teiler (außer der Zahl selber) ist. Beispielsweise ist 28 perfekt, denn $28 = 1 + 2 + 4 + 7 + 14$. Schreiben Sie ein Programm, das für die Zahlen 1 . . . 10000 überprüft, ob diese perfekt sind. Die perfekten Zahlen sollen dabei auf der Konsole ausgegeben werden. *Hinweis:* schreiben Sie zunächst ein Programm, das überprüft, ob eine feste Zahl n perfekt ist.

Aufgabe 4 (5 Punkte). Schreiben Sie ein Programm, das den Nutzer zum Raten einer zufälligen Zahl zwischen 1 und 1000 auffordert, die jedes Mal zu Beginn des Programms generiert wird. Das Programm soll laufen, solange die Zahl nicht geraten wurde. Das Programm soll ausgeben, ob die zuletzt geratene Zahl zu hoch oder zu niedrig war. Eine zufällige `int`-Zahl erhalten Sie, in dem Sie `#include <random>` und `#include <time.h>` einbinden und dann via

```
1     srand(time(NULL)); // Muss nur ein Mal aufgerufen werden
2     int i = rand();
```

¹Sie müssen/sollen hier nichts programmieren.

2

in der `main()`-Funktion. Die Zahl kann mit dem `%`-Operator in den richtigen Bereich transformiert werden.

Achten Sie darauf, dass das Programm sinnvoll mit dem/der Nutzer:in kommuniziert, also zur Eingabe von Zahlen auffordert etc.