

## Einführung in die Programmierung

### Übungsblatt 2

#### Hinweise

- Erstellen Sie bitte eine schriftliche Lösung der nachfolgenden Aufgaben und erzeugen Sie daraus eine pdf-Datei. Sie können auch Ihre (lesbare!) handschriftliche Lösung als pdf einscannen. Achten Sie bitte darauf, dass am Beginn der ersten Seite des pdfs Ihr *Name*, Ihre *Matrikelnummer* und der Name Ihres *Tutors* (Laborübungsgruppe) vermerkt ist. Jede Aufgabe, die durch **(Laborübung)** gekennzeichnet ist, beinhaltet als Lösung auch einen C-Quelltext. Bitte packen Sie Ihr pdf gemeinsam mit allen C-Quelltextdateien in eine zip-Datei, die Sie termingemäß im moodle hochladen (Maximalgröße: 5MB).
- Sie bekommen eine Bewertung Ihrer Lösung mit der Information, ob sie ausreichend zum Bestehen des Übungsblattes ist. Im Falle einer nichtausreichenden Lösung haben Sie Gelegenheit, eine verbesserte Version zeitnah, aber spätestens bis 01.02.16 per E-Mail an Ihren Tutor nachzureichen. Plagiate werden nicht als ausreichende Lösungen anerkannt.

#### Aufgabe 1

Welche der folgenden 12 Zeichenketten sind in der Programmiersprache C zulässige Bezeichner? Geben Sie bei den unzulässigen Zeichenfolgen bitte eine kurze Begründung an!

Deutsch-Englisch	H2O	_1a
main	dec2bin	4fach
Björn	Vergissmeinnicht	tick, trick_und_track
do	übung	WS_15_16

#### Aufgabe 2

- Was versteht man unter einem *Typecast*?
- Nennen und erläutern Sie drei Beispiele für sinnvollen Einsatz von *Typecast*.

**(Laborübung)** Welche Rechenergebnisse liefern die folgenden 9 Ausdrücke jeweils für *x*?

```
int i=4, j=5;
double x;
```

<code>x = (double) i / j;</code>	<code>x = 1.0 * i / j * 10;</code>	<code>x = (10.0 * i) / j;</code>
<code>x = i / j * 10;</code>	<code>x = 1.0 * (i / j) * 10;</code>	<code>x = i / 0.1 * j;</code>
<code>x = i * 10 / j;</code>	<code>x = 10.0 * i / j;</code>	<code>x = i / (0.1 * j);</code>

### Aufgabe 3

Wandeln Sie folgende Zahlen in das jeweils angegebene Zahlensystem um:

$$10010011_{(2)} \longrightarrow ?_{(10)}$$

$$ABCD_{(16)} \longrightarrow ?_{(10)}$$

$$42_{(10)} \longrightarrow ?_{(2)}$$

$$65535_{(10)} \longrightarrow ?_{(16)}$$

### Aufgabe 4

- Geben Sie für die Dezimalzahlen  $-1$ ,  $0$ ,  $-128$ ,  $25$  und  $-92$  die 8-Bit-Darstellung im Zweierkomplement an.
- Die folgenden Bitketten repräsentieren ganze Zahlen im 8-Bit-Zweierkomplement. Welche Dezimalzahlen werden dargestellt?

01111111

10000001

11100101

11000000

### Aufgabe 5

**(Laborübung)** Schreiben Sie ein C-Programm, das vom Benutzer zunächst Geburtstag sowie Geburtsmonat abfragt und dann den Namen des Tierkreiszeichens ausgibt, in dessen Zeitraum das eingegebene Datum liegt (siehe <http://de.wikipedia.org/wiki/Tierkreiszeichen>).

### Aufgabe 6

**(Laborübung)** Das folgende C-Programm enthält einige häufig begangene Fehler. Compilieren Sie das Programm und versuchen Sie, die Bedeutung sowie die Ursache der Fehlermeldungen zu verstehen. Korrigieren Sie das Programm, so dass es fehlerfrei ausgeführt werden kann und die korrekten Ergebnisse liefert.

```
int main(void)
{
    /* Vereinbarung */
    double a, b, c, d, e, f;

    a=4;
    b=2;
    c=2;
    d=1;
    e=2;
    f=10;
    /* Berechnung der Determinante */
    det = (ad)-(bc);
    if(det = 0){
        printf("Berechnung nicht moeglich!\n");
        return -1;
    }
    /* Berechnung von x und y */
    printf("Loesung : \n")
    x = e*d-f*b/det;
    printf("x = %d\n",x);
    y = f*a-e*c/det;
    printf("y = %lf\n",y);
    return 0;
}
```

#### Gewünschte Bedeutung:

Das Programm berechnet die Lösung  $(x, y)$  des linearen Gleichungssystems

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned}$$

und gibt die Ergebnisse für  $x$  und  $y$  aus. Die Werte für  $a, b, c, d, e, f$  sind in diesem Programm fest kodiert.

Das Programm sollte das Ergebnis „Berechnung nicht möglich!“ liefern.

Verändern Sie anschließend die Zuweisung für die Variable  $c$

$c=1$

und führen Sie das Programm wiederholt aus. In diesem Fall sollte das Ergebnis

$$\begin{aligned} x &= -9 \\ y &= 19 \end{aligned}$$

angezeigt werden.