

Übung zur Vorlesung 'Programmieren in C'

Übungsblatt 2

Prof. Thomas Scheffler, Beuth-Hochschule Berlin, FB VII

5. März 2011

Aufgabe 1 Schauen Sie sich den folgenden Programmcode an, in dem einige Fehler enthalten sind:

```
#include <stdio>

int Main()
{
    float celsius = 5,0;
    PRINTF("\n%.1f Grad Celsius entsprechen ", celsius);
    PRINTF("%.1f Grad Fahrenheit.\n", celsius*9/5+32);
}
```

- Schreiben Sie die gefundenen Fehler und eine Erklärung für diese Fehler auf.
Versuchen Sie die Fehler selbst (ohne die Hilfe des Compilers) zu finden!
- Überprüfen Sie Ihr korrigiertes Programm mit dem Compiler.
Haben Sie alle Fehler selbst gefunden? Welche Fehler haben Sie nicht gefunden? Erklären Sie die Fehler die in dem Programm enthalten waren!

Bitte geben Sie in allen Aufgaben die erforderliche Erklärungen mit an!

Sie können diese entweder als Kommentar im Programm kennzeichnen oder als sepeparate Textdatei einreichen. Kennzeichnen Sie erstellte Programme mit Ihrem Namen. Achten Sie auf die korrekte Formatierung und fügen Sie zu Ihren Programmen Kommentare hinzu.

Aufgabe 2 In der folgenden Aufgabe sollen Sie besser verstehen lernen wie der Compiler mit Zahlen umgeht und welche Werte für verschiedene Ausdrücke gebildet werden.

- Geben Sie die erwarteten Ergebnisse zuerst ohne die Hilfe des Compilers an. Erklären Sie die Ausgabe!
- Überprüfen Sie Ihre Ergebnisse und Begründen Sie eventuell festgestellte Abweichungen!

- a. Welchen Wert hat x?

```
int x = 2.5 * 3.0;
```

- b. Welchen Wert hat x?

```
int x = 16 / 010;
```

- c. Was wird ausgegeben?

```
int wert = 32;
printf("%x", wert);
```

- d. Was wird ausgegeben?

```
printf("%.2f", 2.5*3.0);
```

Wie unterscheidet sich die Ausgabe von der folgenden Anweisung?

```
printf("%f.2", 2.5*3.0);
```

Aufgabe 3

Erstellen Sie ein neues Programm mit dem Namen `MyDate.c`. Dieses Programm soll ein vorgegebenes Datum in der amerikanischen und in der europäischen Schreibweise ausgeben. Das Datum kann über Variablen in ihrem Programm fest definiert werden.

Um Ihnen den Einstieg in die Programmierung zu erleichtern, können Sie nach der folgenden Anleitung vorgehen:

- Benutzen Sie die Struktur des `Hello World` Programms und stellen Sie sicher das Sie dieses kompilieren und ausführen können.
- Definieren Sie in dem Programm die folgenden Variablen: `day`, `month` und `year`.
`day` enthält den Tag des Monats, `month` den Monat und `year` das Jahr. Von welchem Typ sind diese Variablen?
Weisen Sie den Variablen Werte zu, welche dem heutigen Datum entsprechen.
- Geben Sie die Werte auf dem Bildschirm aus. Stellen Sie jeden Wert auf einer eigenen Bildschirmzeile dar. Das ist ein Zwischenschritt der ihnen dabei hilft zu überprüfen, ob das Programm funktionsfähig ist.
- Verändern Sie das Programm weiter, so dass es als nächstes das Datum im amerikanischen Standardformat darstellt: `mm/dd/yyyy`.
- Passen Sie ihr Programm erneut an, um eine Ausgabe nach folgendem Muster zu erzeugen, laden Sie das fertige Programm in Moodle hoch:

```
American format:
3/18/2009
European format:
18.3.2009
```

Diese Übung soll Ihnen dabei helfen, formatierte Ausgaben von Werten unterschiedlicher Datentypen mittels der `printf()` Funktion zu erzeugen. Weiterhin sollen Sie die kontinuierliche Entwicklung von komplexen Programmen durch das schrittweise Hinzufügen von einigen, wenigen Anweisungen erlernen.

Aufgabe 4

In dieser Übung werden wir das Programm aus Übung 3 anpassen und verändern, so die Ausgabe nicht direkt im Hauptprogramm erfolgt, sondern dass die Ausgabe innerhalb einer eigenen Funktion erfolgt.

Die Funktionalität des Programms bleibt dabei gleich. Wir strukturieren es aber neu, so dass es aus einem Hauptprogramm und mehreren Unterprogrammen oder Prozeduren entstehen. Dieses Prinzip nennt man auch *strukturierte Programmierung*.

- a. Schreiben Sie eine Funktion mit dem Namen `PrintDateAmerican`. Diese Funktion hat die Parameter `day`, `month` und `year` und gibt das Datum im amerikanischen Standardformat aus.
- b. Testen Sie die Funktion indem Sie die Funktion aus dem Hauptprogramm heraus aufrufen und die entsprechenden Werte als Argumente übergeben. Ihre `main()` Funktion sollte in etwa folgendermaßen aussehen:

```
int main()
{
    PrintDateAmerican(22, 4, 2010);
}
```

- c. Nachdem Sie die Funktion `PrintDateAmerican` erfolgreich erstellt und ausgeführt haben, schreiben Sie eine weitere Funktion `PrintDateEuropean` welche über die gleichen Parameter verfügt und das Datum im europäischen Format ausgibt.

Das Ergebnis der Ausgabe sollte dem Muster aus Aufgabe 3 entsprechen.

Laden Sie die Quellcode-Dateien (*.c) für alle Aufgaben und eventuell vorhandene Erklärungen in Moodle hoch.