

## Arbeitsblatt: INF1

Name:

Kurznamen:

## Einfache Ein-/Ausgabe

### Ziele

- Ein weiteres Beispielprogramm testen und verstehen
- Entwicklungsumgebung für C auf dem Notebook

### Einführung

Nach dem letzten Praktikum sollten Sie C-Programme erstellen und übersetzen können. In diesem Praktikum sollten Sie sich noch ein wenig mit der Kommandozeile auseinandersetzen und auch Eingaben einlesen können.

### Aufgabe: Hello printf

Zur Erinnerung hier noch einmal das erste Programm des letzten Praktikums:

```
/* Programm hello.c */  
  
#include <stdio.h>  
  
int main (int argc, char** argv) {  
    printf("Hello INE1\n\n");  
    return 0;  
}
```

Die Funktion *printf* aus der *stdio*-Bibliothek dient zur Ausgabe von Texten (auch als Strings oder Zeichenketten bezeichnet) auf der Konsole.

Neben der einfachen Ausgabe einer Zeichenkette, können auch z.B. numerische Werte (durch ",", abgetrennt) ausgegeben werden.

```
printf(format, arg2, arg3, ...);
```

Das erste Argument ist speziell: Es enthält den *Formatstring* für die Ausgabe. Die weiteren Argumente dienen dazu, an bestimmten Stellen in den String *eingesetzt* zu werden. Je nach Datentyp muss der Formatstring dazu bestimmte Kontrollsequenzen enthalten. Diese beginnen jeweils mit eine "%".

```
%d    int (oder %i): ganze Zahl (ohne Kommastellen)  
%f    float/double (Zahl, die Nachkommastellen enthalten kann)  
%lf   double (nur bei scanf: Zahl, die Nachkommastellen enthalten kann)
```

%s    **Zeichenkette (=String)** ausgeben  
%x    **int** als Hexadezimalzahl ausgeben  
%%    **Prozentzeichen** ausgeben

Mehr zu *printf* finden Sie unter folgender URL

<http://www.cplusplus.com/reference/clibrary/cstdio/printf/>

Überlegen Sie zunächst, wie die Ausgabe aussehen wird. Probieren Sie aus, ob Ihre Überlegung stimmt und tragen Sie die Resultate ein:

```
double dValue = 2/3.0;  
int iValue = 3;
```

```
printf("%d\n", 17);
```

a)

```
printf("%d**%d%%\n", 17, 24);
```

b)

```
printf("%x\n", 21*2);
```

c)

```
printf("%d\n", 3/4);
```

d)

```
printf("%f\n", 3/4);
```

e)

```
printf("%f\n", 3.0/4.0);
```

f)

```
printf("%.4f\n", iValue+1.0/3.0);
```

g)

```
printf("%.4f\n", iValue+2.0/3.0);
```

h)

```
printf("%.4f\n", 2/3.0+1.0/3.0);
```

i)

```
printf("%.f\n", dValue+1.0/3.0);
```

j)

```
printf("%s bekommt Note %d\n", "Max", iValue*2);
```

k)

## Abgabe

Praktikum: INF2

Filename: INF2.pdf

## Fahrenheit zu Celsius und zurück

Gegeben sei folgendes Programm

```
/**  
 * Nach Anforderung einer Temperatur in Fahrenheit wird die Temperatur  
 * in Celcius berechnet und ausgegeben.  
 * @author Gerrit Burkert  
 * @version 17-jul-2012  
 */  
#include <stdio.h>  
int main (int argc, char** argv) {  
    float celcius, fahrenheit;  
    // Eingabe in Fahrenheit
```

```

    printf("Temperatur in Fahrenheit eingeben: ");
    scanf("%f", &fahrenheit);
    // Ausgabe in Celcius
    celcius = 5.0f * (fahrenheit - 32.0f) / 9.0f;
    printf("Temperatur in Celcius: %f\n", celcius);
    return 0;
}

```

Es wird u.a. die Funktion `scanf` verwendet. Diese liest einen Wert von der Kommandozeile ein und speichert diese in Variablen. Wie beim `printf` wird das Format bzw. der Datentyp durch einen Formatstring als erster Parameter definiert. Der Variablen, in der der eingelesene Wert gespeichert werden soll, muss ein "&" vorangestellt werden (wieso wird später erklärt). Bei `double` Variablen muss `"lf"` als Formatstring angegeben werden.

## Aufgabe 1

Übersetzen Sie das obige Programm und führen Sie es aus; schreiben Sie weiter ein **Fahrenheit** Programm so um, dass statt dem Datentyp `float` der wesentlich genauere Datentyp `double` verwendet wird.

### Hinweise

- Ihre Main Funktion muss die *Signatur* `int main(int argc, char** argv)` besitzen.
- Bei `scanf` muss als Formatstring `"%lf"` für `double` angegeben werden.
- Damit Eingaben in NetBeans gemacht werden können, muss in den Projekt Properties (über Rechtsklick auf das Projekt → Project Properties erreichbar) unter Run → Console Type ein **Externes Terminal** angegeben werden (siehe auch Installationsanleitung)

## Abgabe 2

Praktikum: INF2.1

Filename: `fahrenheit.c`

## Aufgabe 2

Schreiben Sie ein neues Programm **Celsius**, das von Celsius nach Fahrenheit rechnet - lösen Sie dafür einfach die Formel im Beispielpogramm nach *Fahrenheit* auf.

## Abgabe

Praktikum: INF2.2

Filename: `celsius.c`

## Aufgabe 3: Compilation und Ausführen auf der Kommandozeile (optional)

Ein C Programm kann auch auf der Kommandozeile (`cmd`) übersetzt und ausgeführt werden. Navigieren Sie Kommandozeilen Eingabe in das Quellenverzeichnis, wo sich `temperature.c` befindet.

Mittels `gcc fahrenheit.c -o fahrenheit.exe` kann das Programm übersetzt werden (im Verzeichnis in dem sich die Quelle befindet). `fahrenheit.exe` kann dann direkt in der Kommandozeile ausgeführt werden.