

1. Übung: Schreiben Sie ein Programm, dass in der zweiten Zeile „Gewerbliche Schule Leutkirch“ und in der vierten Zeile „ITG11“ auf dem Bildschirm ausgibt.

Variablen:

Bevor eine Variable verwendet werden, kann muss sie deklariert werden. Die wichtigsten Basistypen sind:

int	: ganze Zahl	(4 Byte)
float	: reelle Zahl(mit Nachkommastellen)	(4 Bytes)
double	: doppelt genaue reelle Zahl	(8 Bytes)
char	: einzelnes Zeichen des Zeichensatzes	(1 Bytes)

Kommentare:

Kommentare werden in */* Kommentar */* geschrieben.

Diese werden vom Compiler ignoriert.

Beispielprogramm(*kommentiert*):

```
#include<stdio.h>
int main()
{
    /*Die Variablen i,j und summe werden */
    /*als ganze Zahlen deklariert*/
    int i;
    int j;
    int summe;

    i = 11;        /*Setze die Variable i gleich 11*/
    j = 12;        /*Setze die Variable j gleich 12*/
    summe = i + j; /*Es wird die Summe aus i + j gebildet*/

    printf("\n %d + %d ist gleich %d \n\n", i , j , summe);

    getchar();

    return 0;
}
```

printf("%d", i);



Mit jedem **%d** (für double) greift die Funktion auf das nächste Argument **i** und setzt die entsprechende Zahl in den ausgegebenen Text ein.

2. Übung: Nun soll das Programm abgeändert werden, dass die Zahlen **i=1,25** und **j=2,75** addiert werden.

a) Welche Änderungen müssen am Quellcode vorgenommen werden?
(Bitte protokollieren Sie diese Änderungen.)

```
scanf("%d", &i);
```

Der Befehl `scanf` dient zum Einlesen einer Variable.

scanf ist das Gegenstück zu `printf`.

Wird die Funktion **scanf** aufgerufen, wartet das Programm so lange bis ein `\n`-Zeichen mit der Enter-Taste eingegeben wird.

Abhängig von der Formatanweisung werden die Zahlen entsprechend umgewandelt.

z.B. durch `%d` erfolgt die Umwandlung in eine ganze Zahl(Dezimal)

`%i` steht für integer

`%f` steht für float

`%e` steht für exponential Darstellung

`%c` für ASCII Zeichen

3. Übung: Schreiben Sie ein Programm, dass **zwei ganze Zahlen** einliest und mit beiden Zahlen die 4 Grundrechnungsarten **+, -, •, :** durchführt. Eingabe und Ausgabe sollen wie folgt aussehen:

Geben Sie die erste Zahl ein:

Geben Sie die zweite Zahl ein:

A + B =

A – B =

A • B =

A / B =

Divisionsrest:

Um den Divisionsrest zu erhalten geben Sie bitte `A % B` ein.

z.B. `9 % 4` ist 1.

4. Übung: Erweitern Sie das obige Programm, damit auch der **Divisionsrest** ausgegeben wird (**für ganze Zahlen d.h. integer**).

Inkrement- und Dekrement-Operatoren:

Es kommt sehr häufig vor, dass man zu einer Variable 1 subtrahieren oder addieren muss. Zu diesem Zweck stehen in C Inkrement- und Dekrement Operatoren zur Verfügung.

5. Übung: Analysieren Sie folgendes Programm.

```
#include <stdio.h>

int main()
{
    int i, j;
    i = 5;
    j = ++i;
    printf("i ist %d, j ist %d\n", i, j);
    i = 5;
    j = i++;
    printf("i ist %d, j ist %d\n", i, j);
    getchar();
    return 0;
}
```

Welche Ausgaben erzeugen die Zuweisungen?

Worin unterscheiden sich die beiden Zuweisungen?

! Bitte mit Begründung !

Weitere Bibliotheksfunktionen:

system („clr“) :löscht den Bildschirm

getchar() :erst weiter im Programm nach Betätigung einer Taste.

_flushall() :löscht Ein-/Ausgabepuffer

(Wichtig bei getchar, damit nichts Altes im Puffer bleibt.)

6. Übung: Fordern Sie den Benutzer nacheinander auf,

- eine ganze Zahl
- eine Fließkommazahl
- eine Exponentialzahl
- ein ASCII - Wert einzugeben.

Danach geben Sie diese Werte aus, mit dem entsprechenden Hinweis, um welche Zahl es sich handelt. Zwischen den Ein- und Ausgaben verwenden Sie zur besseren Übersichtlichkeit jeweils eine neue Zeile.

Versehen Sie Ihr Programm mit erläuternden Kommentaren.