 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Formatierte Ausgabe mit cout	Seite 1 von 6

In der Programmiersprache C++ steht zur verbesserten Ausgabe auf der Konsole die Klasse **cout** (der so genannte output stream) zur Verfügung (deklariert in **iostream**). Die auszugebenden Werte werden mit dem <<-Operator an cout übergeben. Für formatierte Ausgaben stehen so genannte **Manipulatoren** und **Flags** zur Verfügung, welche beispielsweise die Stellenzahl, rechts-/linksbündige Ausgabe, Anzahl der Kommastellen usw. festlegen. Um alle Manipulatoren nutzen zu können muss die Header-Datei **iomanip** eingebunden werden.

### Formatierte Ausgabe von Texten:


Das folgende Beispiel deklariert und initialisiert ein Datenfeld für drei Texte. Als Manipulator wird ausschließlich **endl** benutzt, welcher einen Zeilenvorschub erzeugt.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    /*Datenfeld für 3 Artikel mit max. je 20 Zeichen einrichten und initialisieren*/
    char artikel[3][20];
    strcpy_s(artikel[0],"Disketten");
    strcpy_s(artikel[1],"Laserdrucker");
    strcpy_s(artikel[2],"Monitor");
    /*Ausgabe der Artikel*/
    for(int i=0;i<3;i++)
    {
        /*unformatierte Ausgabe mit abschließendem Zeilenvorschub*/
        cout<<artikel[i]<<endl;
    }
    _getch();
    return(0);
}
```

Mit dem Manipulator **setw( )** lässt sich die gewünschte Stellenzahl einstellen. Dem Manipulator wird die Stellenzahl übergeben. Zum Beispiel: **cout<<setw(20)<<...** – Die Ausgabe erfolgt per Voreinstellung rechtsbündig; die vorlaufenden Stellen werden mit SPACE-Zeichen (Füllzeichen) aufgefüllt. Die Festlegung erfolgt nur für die aktuelle cout-Anweisung und wird anschließend wieder auf den Standardwert zurück gesetzt.

Die Ausgabeanweisung in vorigem Beispiel wird verändert:

```
...
//Ausgabe der Artikel
//Ausgabe mit 20 Stellen, standardmäßig rechtsbündig mit SPACE als Füllzeichen
for(int i=0;i<3;i++)
{
    cout<<setw(20)<<artikel[i]<<endl;
}
...
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Formatierte Ausgabe mit cout	Seite 2 von 6

Die Ausgabe:

```

Disketten
Laserdrucker
Monitor
```

Mit dem Manipulator `setfill()` lässt sich das Füllzeichen definieren. Zum Beispiel `cout<<setfill(,.' );` legt den Punkt als Füllzeichen fest. Im Gegensatz zu `setw()` bleibt diese Einstellung für alle folgenden cout-Anweisungen erhalten.

Die Steuerung der links-/oder rechtsbündigen Ausgabe lässt sich über so genannte **Flags** steuern. Ein Flag ist ein Merkmal, welches entweder gesetzt oder nicht gesetzt ist. Die Flags für `cout` sind in einer Klasse `ios` definiert. In dieser Klasse befinden sich u.a. die Flags `left` und `right`. Der Zugriff auf diese Flags erfolgt unter Angabe des Klassennamens mit dem Sichtbarkeitsoperator `::` (z.B. `ios::left`). Die Flags werden beeinflusst mit dem Manipulator `setiosflags(.`

Zum Beispiel: `cout<<setiosflags(ios::left);` legt dauerhaft eine linksbündige Ausgabe fest bis die Einstellung geändert wird. Mit `cout<<resetiosflags(ios::left)` wird die Einstellung wieder rückgängig gemacht, das heißt wieder rechtsbündige Ausgabe.

In folgendem Beispiel erfolgt die Ausgabe der Artikel linksbündig und mit dem Punkt als Füllzeichen:

```

#include<iostream>
#include<iomanip>
#include<conio.h>
using namespace std;
int main()
{
    //Datenfeld für 3 Artikel mit max. je 20 Zeichen einrichten und initialisieren
    char artikel[3][20];
    strcpy_s(artikel[0],"Disketten");
    strcpy_s(artikel[1],"Laserdrucker");
    strcpy_s(artikel[2],"Monitor");
    //Ausgabe der Artikel
    //Ausgabe mit 20 Stellen linksbündig und '.' als Füllzeichen
    cout<<setiosflags(ios::left)<<setfill('.');
    for(int i=0;i<3;i++)
    {
        cout<<setw(20)<<artikel[i]<<endl;
    }
    _getch();
    return(0);
}
```

Die Ausgabe:

```

Disketten.....
Laserdrucker.....
Monitor.....
```


### Formatierte Ausgabe von ganzen Zahlen:

Das folgende Beispiel legt ein Datenfeld mit 4 ganzen Zahlen an und initialisiert dieses. Anschließend wird das Datenfeld mit unterschiedlichen Formatierungsangaben ausgegeben. Die Festlegung der Breite und des Füllzeichens erfolgt analog zur Textausgabe mit den Manipulatoren `setw()` und `setfill()`. Mit dem Manipulator `hex` können Zahlen auch im Hexadezimalen Zahlenformat ausgegeben werden (zum Beispiel: `cout<<hex;`)

```
#include<iostream>
#include<iomanip>
#include<conio.h>
using namespace std;
int main()
{
    //Datenfeld für 4 ganze Zahlen einrichten und initialisieren
    int zahl[4]={3,22,412,1024};
    int i;
    //Ausgabe standardmäßig rechtsbündig
    for(i=0;i<4;i++)
    {
        cout<<setw(4)<<zahl[i]<<" "; //Ausgabe mit je 4 Stellen
    }
    cout<<endl;
    //Ausgabe mit führenden Nullen standardmäßig rechtsbündig
    cout<<setfill('0');
    for(i=0;i<4;i++)
    {
        cout<<setw(4)<<zahl[i]<<" "; //Ausgabe mit je 4 Stellen
    }
    cout<<endl;
    //Ausgabe in HEX mit führenden Nullen
    cout<<hex<<setfill('0');
    for(i=0;i<4;i++)
    {
        cout<<setw(4)<<zahl[i]<<" "; //Ausgabe mit je 4 Stellen
    }
    cout<<endl;
    _getch();
    return(0);
}
```

### Die Ausgabe:

```
3    22   412 1024
0003 0022 0412 1024
0003 0016 019c 0400
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Formatierte Ausgabe mit cout	Seite 4 von 6

### Formatierte Ausgabe von Fließkommazahlen:

Fließkommazahlen lassen sich entweder im Festkommaformat oder im wissenschaftlichen Format (evtl. mit Zehnerpotenzen) darstellen. Mit den Flags `ios::fixed` bzw. `ios::scientific` lässt sich das Ausgabeformat einstellen.

Der Manipulator `setprecision()` legt die Anzahl der Nachkommastellen fest.

#### Zum Beispiel:

```
cout<<setiosflags(ios::fixed)<<setprecision(2);
```

Ausgabe im Festkommaformat mit zwei Nachkommastellen. Die folgenden Stellen werden gerundet.

```
#include<iostream>
#include<iomanip>
#include<conio.h>
using namespace std;
int main()
{
    //Datenfeld für 4 Fließkommazahlen einrichten und initialisieren
    double zahl[4]={5.0,3.1415,0.025,100.4567};
    int i;
    //Ausgabe mit Voreinstellungen
    for(i=0;i<4;i++)
    {
        cout<<zahl[i]<<endl;
    }
    cout<<endl;
    //Ausgabe mit 8 Stellen, standardmäßig rechtsbündig
    for(i=0;i<4;i++)
    {
        cout<<setw(8)<<zahl[i]<<endl;
    }
    cout<<endl;
    //Ausgabe in Festkommadarstellung und zwei Nachkommastellen
    cout<<setiosflags(ios::fixed)<<setprecision(2);
    for(i=0;i<4;i++)
    {
        cout<<setw(8)<<zahl[i]<<endl;
    }
    cout<<endl;
    //Ausgabe im wissenschaftlichen Format mit 3 gültigen Stellen
    cout<<scientific<<setprecision(3);
    for(i=0;i<4;i++)
    {
        cout<<setw(8)<<zahl[i]<<endl;
    }
    cout<<endl;
    _getch();
    return(0);
}
```

### Die Ausgabe:

```

5
3.1415
0.025
100.457

    5
    3.1415
    0.025
    100.457

        5.00
        3.14
        0.03
        100.46

5.000e+000
3.142e+000
2.500e-002
1.005e+00

```

### Übung:

Das folgende Programm erfasst Datensätze für die Anzahl, den Namen und den (Netto-)Preis für verkaufte Artikel und speichert diese in den entsprechenden Datenfeldern.

```

#include<iostream>
#include<iomanip>
#include<conio.h>
using namespace std;
int main()
{
    int anzahl[20];
    char name[20][20];
    double preis[20]; //Das ist der Nettopreis
    double summe=0;
    int i=0;
    do
    {
        cout<<"\n"<<i+1<<". Datensatz:"<<endl;
        cout<<"Anzahl: ";cin>>anzahl[i];
        cout<<"Name: ";cin>>name[i];
        cout<<"Preis: ";cin>>preis[i];
        cout<<"'ESC' = Ende ... "<<flush;
        i++;
    }
    while(!_getch()!=0x1b);
    system("cls"); //Bildschirm löschen
    /**** Hier die Ausgabefunktion für die Rechnungsschreibung einfügen!!! *****/
    return(0);
}

```

Nach der Eingabe der Datensätze und dem Löschen des Bildschirms soll die Ansicht einer Rechnung ausgegeben werden. Alle Zahlenangaben sind rechtsbündig auszugeben und Preise mit 2 Nachkommastellen. Der Text ist linksbündig auszugeben mit Punkt als Füllzeichen. Zu berechnen sind außerdem Gesamtnettopreis, Mehrwertsteuer und Gesamtbruttopreis. Zur Ausgabe ist ausschließlich **cout** mit entsprechenden Manipulatoren zu verwenden!

Der Bildschirm sollte etwa folgendermaßen aussehen:

```
***** RECHNUNG *****
Anzahl Artikelname           Preis    Gesamt
-----
120 CD.....                2.10    252.00
10 Drucker.....            49.50   495.00
1 Monitor.....             198.00  198.00
                                     _____
                                Gesamt(netto):  945.00
                                19% MwSt:    179.55
                                     _____
                                Gesamt(brutto): 1124.55
                                     _____
```