

## II. 4 Variablentypen

---

### II. 4. 1 Fließkommavariablen

Es stehen drei Typen von Fließkommavariablen zur Verfügung. In der Tabelle sind die Mindestgrößen angegeben.

### Typumwandlung bei gemischten Ausdrücken

#### Übung:

Gibt es einen Unterschied zwischen

```
int n=5;  
cout << 1/n << endl;
```

und

```
int n=5;  
cout << 1.0/n << endl;
```

?

## Typumwandlung

---

Werden zwei Variablen unterschiedlichen Typs durch einen Operator miteinander verknüpft, dann bekommt das Ergebnis den genaueren Typ des Ergebnisses.

Bei einer Zuweisung wird das Ergebnis auf der rechten Seite stets in den Datentyp der linken Seite umgewandelt.

Probieren Sie mehrere Typen für `erg` aus (für beide Fälle `1/n` und `1.0/n`) :

```
typ erg;  
int n=5;  
  
erg = 1/n;  
cout << erg << endl;
```

Erstellen Sie eine Tabelle mit den Lösungen.

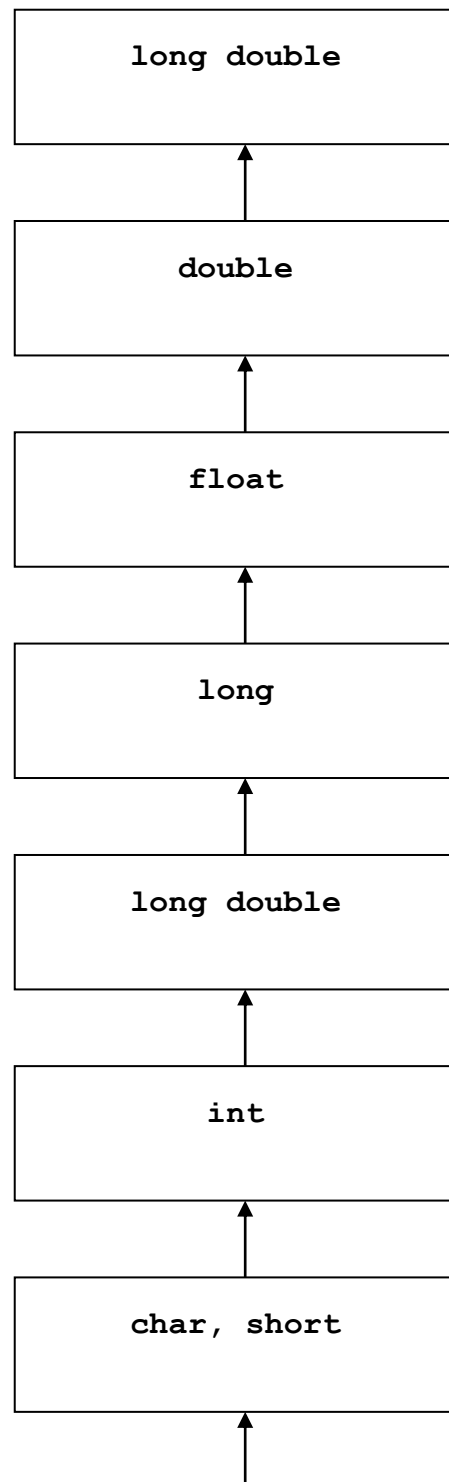
typ von erg	1/n	1.0/n
int		
float		

Wie sieht die Rangfolge der automatischen Typumwandlung des Compilers aus?

# Typumwandlung

---

Rangfolge der automatischen Typumwandlung:



## cast-Operator

---

Mit dem `cast`-Operator können wir temporär eine Typumwandlung vornehmen. Die Schreibweise ist einfach

```
(typ) variable
```

oder übersichtlicher

```
(typ) (variable)
```

### Übung:

Was sind die Unterschiede von

```
int a=3, b=2  
float c;
```

```
c=a / b;
```

```
int a=3, b=2  
float c;
```

```
c=(float) (a) / (float) (b);
```

**Achtung:** Nach der Typumwandlung sind die Variablen `a` und `b` noch immer Ganzzahlen.

Die Typumwandlung mit dem `cast`-Operator bezieht sich nur auf den Wert, den die Variable während der Verknüpfung repräsentiert.

Zur Übersichtlichkeit, insbesondere bei langen Programmen, sollte man den Typumwandlungsoperator auch dann benutzen, wenn es nicht unbedingt nötig ist. Man kann dann sofort erkennen, daß die Variablentypen unterschiedlich sind.

Beispiel: `int a=4; float b;`

```
b=a;
```

oder besser

```
b=(float) (a);
```

## **cmath**

---

Mit der Headerdatei `cmath` stehen Ihnen die gebräuchlichsten mathematischen Funktionen für Fließkommazahlen zur Verfügung.

Eingebunden wird `cmath` mit dem Präprozessorbefehl

```
#include <cmath>
```

Als Kopie folgt eine Liste der Funktionen mit kurzen Erklärungen.

Die Umrechnung eines Logarithmus zu einer beliebigen Basis  $b$  nehmen Sie nur zur Kenntnis. Wichtig ist der kleine Kapitel zur Umrechnung von Winkel- in Bogenmaß und umgekehrt.

Am Ende der Kopien sind dann noch einige Manipulatoren ( aus `<iomanip>` )aufgeführt:

- `showpoint`
- `noshowpoint`
- `scientific`
- `fixed`
- `setprecision()`

## Übungen: cmath

---

### Aufgaben:

1.) Berechnen Sie die Absolutwerte, d.h. die Beträge, von

```
int a=4, b=5, c=4.5, d=-2, e=0.0;  
float f=3.0, g=-3.4, cos=-3.9, sin=-15;
```

2.) Berechnen Sie den Sinus und den Cosinus von 45°.

3.) Berechnen Sie "5 hoch 4".

4.) Berechnen Sie "2 hoch 16 und davon die Wurzel".

5.) Berechnen Sie "e hoch 2,5".

6.) Gibt es 15 % 3.5 ? Was ist das Ergebnis?

## II. 4. 2 char

---

Der Variablentyp `char` (für engl. character = Zeichen) ermöglicht das Speichern eines Zeichens. Die Benutzung ist einfach:

```
int main()
{
    char x,y;

    x='A';
    cout << "Zeichen eingeben: " ;
    cin >> y;
    cout << "Zeichen war :" << y << endl;
}
```

### **Vorsicht:**

Zeichenkonstanten stehen zwischen einfachen Anführungsstrichen.

Der Bereich von `char`-Variablen geht von 0 bis 255. Der Compiler schreibt in die Variable nicht das 'A' selbst, sondern den Wert, mit dem der Buchstabe 'A' in der ASCII-Tabelle codiert ist. Es gibt aber auch `signed char` mit einem Wertebereich von -128 bis 127.

Die `char`-Variablen lassen sich wie ganzzahlige Variablen behandeln. Mit Hilfe des `cast`-Operators kann der Typ gewandelt werden, z.B.

```
char x = 'P';
cout << "Wert von " << x << " ist " << (int) (x);
```

Der Wert ist ?

Was ist das Ergebnis, wenn in eine `float`-Variable umgewandelt wird ?

## char

---

### Übung:

Was gibt folgendes Programm aus:

```
int main()
{
    int x;

    for (x=0;x<256;x++)
    {
        cout << " Das Zeichen " << (char) (x);
        cout << " hat den Wert " << x << endl;
    }
}
```

Hier wird eine `int`-Variable als `char` ausgegeben.

Mit Zeichenkonstanten kann auch gerechnet werden. Ein Beispiel:

```
int x;
x=15 + 'n';
cout << x << endl;
x='A' + 'C';
cout << x << endl;
x=2*'A';
cout << x << endl;
```



## char-Funktionen

---

char-Variablen können auch in der switch-case-Anweisung benutzt werden.  
Probieren Sie

```
switch (z)
{
    case 'm': cout << " m wurde gewählt" << endl;
              break;

    case 'l': cout << " l wurde gewählt" << endl;
              break;

    default:  cout << "Mist" << endl;
}
}
```

**Funktionen für char-Variable:**

## char-Funktionen

---

Beispiel:

```
#include <iostream>
#include <cctype>

int main()
{
    char w;

    do
    {
        cout << " Bitte eine Ziffer eingeben, ";
        cout << "keinen Buchstaben:";
        cin >> w;
    } while (isalpha(w));
    cout << " Ziffer " << w << " eingegeben";
}
```

In welchem Verzeichnis liegen die Dateien "iostream" und "cctype" ?

Als Kopien liegen kurze Erläuterungen zu den Funktionen bei.

# Übung

---

Entwickeln Sie ein Flugbuchungs-Entscheidungsprogramm für Ihre nächste Pauschalreise für folgende Situation:

- 3 Anbieter fliegen zu Ihrem gewünschten Ort und Hotel
- die Preise und die Abflug-Flughäfen sind unterschiedlich
- einen Fahrkilometer zum Flughafen kalkulieren Sie mit 0,45 DM.

Das Programm soll Sie jeweils nach Flughafen, Preis und Entfernung des Flughafens fragen. Geben Sie die Daten interaktiv ein und ermitteln Sie die gesamten Kosten (Flug + Pkw) der Reise. Diese stellen Sie für alle 3 Fälle dar.

Entscheiden Sie sich je nach Neigung für einen Anbieter, der als "A", "B" oder "C" abgefragt werden soll. Die Ausgaben vor Ort belaufen sich auf ungefähr 75% des Pauschalpreises (also ohne die Pkw-Kosten). Geben Sie anschließend die geschätzten Gesamtkosten aus.

- a.) Entwickeln Sie ein Grob-Struktogramm.
- b.) Entwickeln Sie ein Struktogramm, das auch die Variablen bezeichnet und die Formeln deutlich macht.
- c.) Schreiben Sie den Programm-Code auf Papier.
- d.) Was sagt Ihr Nachbar? Ist das Programm lauffähig?

## II. 5 Variablenfelder (Arrays)

---

Stellen Sie sich vor, Sie wollten z.B. 15 Zahlen in mit Hilfe verschiedener Variablen speichern. Mit Variablennamen wie `a, b, c, d`, dürfte Ihnen bei einer noch größeren Anzahl bald "die Puste ausgehen". Hier helfen Felder (engl. Arrays).

Ein Array wird folgendermaßen definiert:

```
typ name [anzahl];
```

Die Anweisung definiert ein Feld namens `name`. Es stellt die Menge von `anzahl` Variablen des Typs `typ` zur Verfügung.

Beispiel:

```
int a[15];
```

Nun gibt es 15 Variable, nämlich `a[0]` bis `a[14]`. **Also Vorsicht !!!**

Der Feldindex beginnt mit 0. Der Index eines arrays mit `n` Elementen reicht von 0 bis `n-1`.

[ Bild mit Schubladen]

Die Zuweisung der einzelnen Feldelemente ist einfach

```
a[0] = 5;  
a[3] = 67;
```

Ist folgendes Beispiel gültig?

```
b=5;  
a[3+b] = b;
```

## array

---

### Übung:

Was führt folgendes Programm aus ?

```
int main()
{
    int x[10], y;

    for (y=0;y<10;y++)
    {
        cout << "Bitte Zahl "<< y+1 << "eingeben :";
        cin >> x[y];
    }

    for(y=0;y<10;y++)
    {
        cout << x[y] << " mal 2 ist " << x[y]*2 << endl;
    }
}
```

Ändern Sie y in `float`; lassen Sie y von 0.1 bis 10.1 laufen.

Ein array kann auch mit der Definition initialisiert werden:

```
int c[6] = {1,3,-5,7,9,11};

float z[4] = {34.56, 24.09, 1.0, 23.2};
```

Diese einfache Wertzuweisung an ein gesamtes (!) Array funktioniert allerdings nur bei der Definition.

# array

---

## Mehrdimensionale Arrays

Eine typische Anwendung eines mehrdimensionalen arrays ist ein zweidimensionales Zahlenfeld, auch Matrix genannt.

Beispiel:

3	5	0	0
-1	0	12	-9
0	2	10	3

Dies ist eine 3 x 4 Matrix, d.h. sie besteht aus 3 Zeilen und 4 Spalten. Soll der Array-Name `k` lauten, so ist die Definition:

```
int k[3] [4];
```

Die Feldelemente sind

```
k[0][0]=3;
k[0][1]=5;
k[0][2]=0;
k[0][3]=0;
k[1][0]=-1;
k[1][1]=0;
k[1][2]=12;
...
```

Auch hier ist bei der Definition die Initialisierung möglich:

```
int k[3] [4] = {    {3,5,0,0},
                    {-1,0,12,-9},
                    {0,2,10,3}
                };
```

## array

---

Intern werden die mehrdimensionalen Felder als eindimensionales Feld verwaltet.  
Die Speicherreihenfolge lautet:

`k[0][0], k[0][1], ... , k[0][3], k[1][0], k[1][1], ...`

### Übung:

Erzeugen Sie ein eindimensionales Feld `b`, dem der Inhalt des Feldes `k` in einer `for`-Schleife übergeben wird.

Was bewirkt folgende Schleife?

```
float a[100];

for(i=0;i<100,i++)
{
    cout << " " << a[i] << " ";
    if ((i+1)%10 == 0)
    {
        cout << endl;
    }
}
```

## Übung

---

Betrachten Sie folgende Abbildung:

Eine Person A findet sich in 50 m Entfernung vom Strand. 100 m weiter am Strand droht eine Person B im Wasser zu ertrinken. Die Entfernung von B zum Strand beträgt 50 m. A möchte B möglichst schnell zu Hilfe eilen. Am Strand kann er mit 5 m/s laufen, im Wasser ist er 2,5 m/s schnell.

Schreiben Sie ein Programm, das den Weg ermittelt, auf dem A schnellstmöglich beim Ertrinkenden B ist.



## Lösungsweg

---

- a.) Belege die Uferstrecke mit dem Lot zu A und B mit einer Koordinatenachse  $x$ , wobei die interessante Strecke  $x$  von 0 bis 100 sei.
- b.) Wie schnell ist A für  $x=0$  m,  $x=100$  m und  $x=50$  m?
- c.) Überlege einen Algorithmus, der die Zeit in Abhängigkeit von  $x$  berechnet.
- d.) Lasse nun  $x$  von 0m bis 100m in Schritten von 1m durchlaufen.
- e.) Schreibe die Ergebnisse in ein array[100] oder [101].
- f.) Gib die alle Ergebnisse auf dem Bildschirm aus.
- g.) Für welches  $x$  ist der Wert am kleinsten und wie lautet er?

1.) Nennen Sie die sechs Vergleichsoperatoren und deren C-Code.

2.) Geben Sie in der Tabelle die Mindestgrößen nach ANSI-Norm und die verwendeten Datengrößen des GNU C++-Compilers für verschiedene Datentypen an.

<b>Datentyp</b>	<b>Mindestgröße/[Bytes]</b>	<b>GNU C++-Compiler/[Bytes]</b>
short		
int		
long		
float		
double		

3.) Welche Speicherklassen von Variablen kennen Sie ?

4.) Was liefert

a.)  $22/5$

b.)  $22.0/5$

c.)  $21\%6$

d.)  $3*5-2/2$

e.) 

```
x=2;
x -=3;
cout << x;      ?
```

5.) Welcher Wert wird für a ausgegeben?

a.)    `a=4;`  
      `a/=2;`  
      `a++;`  
      `a=a+a;`  
      `cout << a;`

b.)    `a=2;`  
      `a-=4+a++;`  
      `a=a+a;`  
      `cout <<a;`

6.) Versuchen Sie die folgenden Zuweisungen so umzuschreiben, daß nur jeweils einer der Operatoren `+=`, `-=`, `*=`, `/=` verwendet wird.

a.)    `a=r*a*t*t;`

b.)    `x=x*(3+y);`

c.)    `a=3*a; a=b*a;`

d.)    `a=a*b; b=b+1;`

7.) Falls Sie Fehler in folgenden Programmen finden, streichen Sie diese bitte an:

a.)  
      `*include <iostream>`  
      `using namespace sstd;`  
  
      `float main()`  
      `{`  
          `Cout << "Hallo!";`  
      `}`

b.)

```
/* Programm erstellt von A. Fenske
#include <iostream>;
int quadrat();
{
    int quadrat;
    if (x<10)
        quadrat=(x*x);
        x=return(quadrat)
    }
}
int main
    int, double wert,ergebnis;
    cout >> "Bitte geben Sie eine Zahl ein:";
    cin << wert
    quadrat(wert);
    cout>>"Quadrat von ">>wert>>"ist ">>ergebnis>>endl;
```

8.) Wie lautet die Rangfolge bei der automatischen Typumwandlung?

9.) Entwickeln Sie ein Programm, das nach der Bildschirmeingabe Ihres zu versteuernden Einkommens die von Ihnen zu zahlende Einkommensteuer berechnet (Stand 1.1.96 EStG, §32a) und ausgibt.

Das zu versteuernde Einkommen (in DM-Beträgen) ist vor den weiteren Berechnungen auf den nächsten durch 54 ohne Rest teilbaren Betrag abzurunden, falls es nicht bereits durch 54 ohne Rest teilbar ist.

Die Einkommenssteuer beträgt in DM für zu versteuernde Einkommen:

1. bis 12095 DM (Grundfreibetrag): 0;
2. von 12096 DM bis 55727 DM:  $(86,63y + 2590)y$ ;
3. von 55728 DM bis 120041 DM:  $(151,91z + 3346)z + 12949$ ;
4. von 120042 DM an:  $0,53x - 22842$ .

x ist das abgerundete zu versteuernde Einkommen,

y ist ein Zehntausendstel des 12042 DM übersteigenden Teils des abgerundeten zu versteuernden Einkommens,

z ist ein Zehntausendstel des 55674 DM übersteigenden Teils des abgerundeten zu versteuernden Einkommens.

a.) Überlegen Sie sich grob ein Lösungsschema.

b.) Entwickeln Sie ein Struktogramm nach Nassi Shneidermann. Benennen Sie die benötigten Variablen.

c.) Schreiben Sie Ihren Code auf Papier auf.