Grundrechenarten

Mit Zahlen können ohne weitere Vorkehrungen mathematische Operationen wie z.B. Addition, Subtraktion, Multiplikation usw. ausgeführt werden. Die folgende Tabelle zeigt die Operatoren für ganze Zahlen und Fließkommazahlen:

Mathematische Operationen

Operator	*Bedeutung*	*Beispiel*
+	Addition	2+3
-	Subtraktion	3-2
*	Multiplikation	2*3
/	Division	2/3
%	Modulo - Divisionsrest bei ganzen Zahlen	5%3
( )	Klammern	2*(3+4)
=	Zuweisung	x=2+3
+=	Kurzform Addition mit Zuweisung	x+=3 entspricht x=x+3
-=	Kurzform Subtraktion mit Zuweisung	x-=3 entspricht x=x-3
*=	Kurzform Multiplikation mit Zuweisung	x*=3 entspricht x=x*3
/=	Kurzform Division mit Zuweisung	x/=3 entspricht x=x/3
++	vorherige/nachfolgende Inkrementierung	++i / i++ entspricht i=i+1
--	vorherige/nachfolgende Dekrementierung	--i / i-- entspricht i=i-1

Eine Berechnung kann beliebig viele Operatoren in einer Zeile enthalten. Dabei gilt grundsätzlich Punkt- vor Strichrechnung. Klammerausdrücke werden jedoch vorrangig berechnet. Berechnungen mit gleichwertigen Operatoren werden von links nach rechts ausgeführt. Beachten Sie, dass bei den Kurzformen das Ausgangsobjekt durch eine implizite Zuweisung verändert wird.

Besondere Beachtung ist dem Zuweisungsoperator '=' zu widmen. Er hat die Form

```
bezeichner=ausdruck;
```

Der links stehende ***bezeichner*** stellt ein deklariertes Objekt dar und der rechts stehende ***ausdruck*** ist ein beliebiger mathematischer Ausdruck, dessen Ergebnis dem links stehenden Objekt zugewiesen wird. Im Gegensatz zu einer mathematischen Gleichung dürfen die Seiten nicht vertauscht werden!

Deshalb sind z.B. folgende Zuweisungen zulässig:

```
i = i + 1  // Der Wert von i wird um 1 erhöht (inkrementiert)
zahl1 = zahl1 * zahl1  // zahl1 bekommt den Wert des Quadrates
```

Es ist möglich Ausdrücke unterschiedlicher Datentypen miteinander zu verknüpfen. Das Ergebnis der Verknüpfung ist immer vom Typ der höheren Genauigkeit. Bei der Addition einer Fließkommazahl vom Typ Double mit einer Integer-Zahl ist das Ergebnis vom Typ Double. Die notwendige Umwandlung eines Ausdrucks in den erforderlichen Typ erfolgt automatisch.

Die Umwandlung eines Ausdruckes in den gewünschten Datentyp kann mit sogenannten 'cast'-Ausdrücken erzwungen werden. Dazu wird der gewünschte Datentyp in Klammern vor dem Ausdruck gesetzt.

Beispiel 1:

(int)3.1 liefert als Ergebnis den Integerwert 3.

Beispiel 2:

```
#include < stdio.h >
#include < conio.h >
int main()
{
      int izahl1=5,izahl2=3;
      double dwert1=2.5,dwert2=5.2;
      printf("\n%lf",dwert1+izahl2);        //Ergbebnis vom Typ Double
      printf("\n%i",(int)dwert2*izahl1);    //Ergebnis vom Typ Integer
      printf("\n%lf",(double)izahl1*izahl2);//Ergebnis-Typ Double
      printf("\n");
      return(0);
}
```

Übung :

Analysieren Sie das folgende Programmbeispiel und bestimmen Sie die Werte der Ausgabe bevor Sie das Programm testen!

```
#include < stdio.h >
#include < conio.h >
main()
{
      int izahl1=5,izahl2=3;
      double dwert1=2.5,dwert2=5.0;
      printf("\n%i",izahl1+izahl2*2);
      printf("\n%i",(izahl1+izahl2)*2);
      printf("\n%lf",dwert1*dwert2+2);
      printf("\n%lf",dwert1*(dwert2+2));
      printf("\n%i",izahl1+=5);
      printf("\n%lf",dwert2/=2);
      printf("\n%i",izahl1%izahl2);
      printf("\n%i",izahl2++);
      printf("\n%i",izahl2+=1);
      printf("\n");
      return(0);
}
```