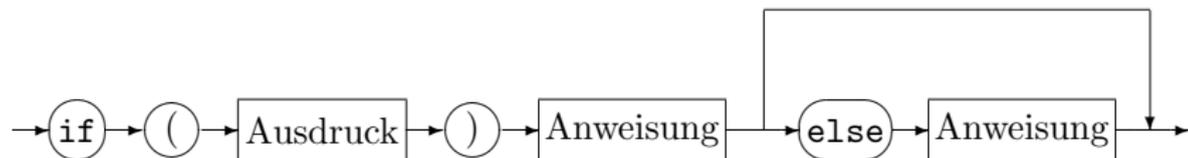


if-Anweisung (vorläufig)



Semantik: *Ausdruck* hat die Bedeutung einer Bedingung
Anweisung(en) werden als "abhängige Anw." bezeichnet

1. Form *Unvollständige if-Anweisung (else-Zweig fehlt)*

Wirkung: Falls *Ausdruck* wahr ist, wird die abhängige *Anweisung* ausgeführt, andernfalls nicht.

2. Form *Vollständige if-Anweisung*

Wirkung: Falls *Ausdruck* wahr ist, wird die erste abh. *Anweisung* ausgeführt, andernfalls die zweite abh. *Anweisung*.

Bei beiden Formen wird danach mit der auf die if-Anweisung folgenden *Anweisung* fortgesetzt.

Häufig: *Bedingung* \equiv *Vergleich*

Anweisung \equiv { *Anw1 Anw2 ...* } (d.h Blockanweisung)

Bsp.: Quadratwurzel

Aufgabenstellung

Ausgabe von x und von \sqrt{x} , falls $x \geq 0$, sonst Fehlermeldung.

Programmfragment

```
double x;
cout << "x: "; cin >> x;
if (x>=0)
    { cout << "x=" << x
        << "  sqrt(x)=" << sqrt(x) << endl; }
else
    { cout << "x=" << x
        << "  sqrt(x) undefiniert" << endl; }
```

Ausgabebeispiele (ohne Einlesevorgang)

```
x=2  sqrt(x)=1.41421
x=-1  sqrt(x) undefiniert
```

Bsp.: Überprüfung von Eingabedaten

Aufgabenstellung

Programmabbruch, falls negative Zahl gelesen

Programmfragment

```
int n;  
cout << "n: "; cin >> n;  
if (n<0) {  
    cout << "n negativ. Abbruch." << endl;  
    exit(1); // Programmabbruch mit Exitcode 1  
}
```

Unterschiede zwischen `return` und `exit`:

<code>return n</code>	Anweisung	Beendet Funktion mit Wert n Datentyp von n : u.a. Zahlen [n fehlt bei Fkt. ohne Ergebnis]
<code>exit(n)</code>	Funktion	Beendet Programm mit Exitcode n Datentyp von n : <code>int</code> (<code>#include <cstdlib></code> notwendig)