

Ein Anweisungsblock in C++
wird

- ☒ ~~A~~ mit { und } begrenzt
- ☐ B nur bei Schleifen erlaubt
- ☒ ~~C~~ auch als Sequenz bezeichnet
- ☐ D nur bei If-Anweisungen eingesetzt

Ein Anweisungsblock in C++
wird

- ☒ ~~A~~ auch als Verbundanweisung bezeichnet
- ☐ B nur bei Schleifen erlaubt
- ☒ ~~C~~ mit { und } begrenzt
- ☐ D nur bei If-Anweisungen eingesetzt

```
if (j > n)
    j = 0;
else
    j = i;
```

A $j=0$ und $j=i$ werden ausgeführt

~~B~~ $j=0$ wird ausgeführt
gegeben $j=5$ und $n=3$

C $j=i$ wird ausgeführt
gegeben $j=5$ und $n=3$

D $j=i$ wird nie ausgeführt

E $j=0$ wenn ($j>n$) den Wert 0 (=false) liefert.

```
if (j > n)
    j = i;
else
    j = 0;
```

A $j=0$ und $j=i$ werden ausgeführt

B $j=0$ wird ausgeführt
gegeben $j=5$ und $n=3$

~~C~~ $j=i$ wird ausgeführt
gegeben $j=5$ und $n=3$

D $j=i$ wird nie ausgeführt

~~E~~ $j=0$ wenn ($j>n$) den Wert 0 (=false) liefert.

Welche Aussagen über logische Ausdrücke sind wahr ?

- ☒ ~~A~~ ein logischer Ausdruck liefert einen booleschen Wert
- ☒ ~~B~~ sie werden als Bedingung für if-Anweisungen eingesetzt
- ☐ C mögliche Werte sind „wahr“, „falsch“ und „undefiniert“
- ☒ ~~D~~ 0 bedeutet „falsch“

Welche Aussagen über logische Ausdrücke sind wahr ?

- ☒ ~~A~~ 0 bedeutet „falsch“
- ☒ ~~B~~ sie werden als Bedingung für Wiederholungsanweisungen eingesetzt
- ☐ C mögliche Werte sind „wahr“, „falsch“ und „undefiniert“
- ☒ ~~D~~ ein logischer Ausdruck liefert einen booleschen Wert

Welche logischen Ausdrücke ergeben 1 (wahr):

~~A~~ 1 && 1

~~B~~ 1 || 0

C 0 && 1

~~D~~ !0

~~E~~ 1 && !0

Welche logischen Ausdrücke ergeben 1 (wahr):

A 1 && 0

~~B~~ 1 || 1

~~C~~ 0 || 1

~~D~~ !0

~~E~~ 1 || !0

In einer Mehrfachverzweigung (switch-Anweisung) ermöglicht das Angeben des Schlüsselworts **case**:

A Die Ausgabe von **case**

~~B~~ Das Ausführen der Anweisungen innerhalb des **case**-Zweiges, wenn die betreffende Konstanten im case-Zweig dem switch-Ausdruck entspricht.

C Die Anzeige eines Faults (=Fehler)

D Die Möglichkeit auf unvorhergesehene Fälle (cases) zu reagieren.

In einer Mehrfachverzweigung (switch-Anweisung) ermöglicht das Angeben des Schlüsselworts **default**:

~~A~~ Das Ausführen der Anweisungen innerhalb des **default**-Zweiges, wenn keine der Konstanten in den case-Zweigen dem switch-Ausdruck entspricht.

B Die Ausgabe von **default**

~~C~~ Die Möglichkeit auf unvorhergesehene Fälle (cases) zu reagieren.

D Die Anzeige eines Faults (=Fehler)

Notieren Sie die Ausgabe des Programms.

```
#include<iostream.h>
int main(){

int x=0;

for(int i=0;i<=4;i=i+2)
    cout << i << (x=x+i);

cout << "x=" << x;
}
```

002246x=6

Notieren Sie die Ausgabe des Programms.

```
#include<iostream.h>
int main(){

int x=2;

for(int i=9;i>=3;i=i-3)
    cout << i << (x=x+i);

cout << "x=" << x;
}
```

911617320x=20

```
#include<iostream.h>

int main(){
    bool logA=1;
    bool logB=0;
    int etc = 1;

    if (logA==etc && !logB)
        cout << etc;

    if (!(etc != 3))
        cout << etc-1;
}
```

Notieren Sie die Ausgabe.

1

```
#include<iostream.h>

int main(){
    bool logA=0;
    bool logB=1;
    int etc = 1;

    if (logA==etc || !logB)
        cout << etc;

    if (!(etc != 1))
        cout << etc-1;
}
```

Notieren Sie die Ausgabe.

0

```
#include<iostream.h>
int main() {
    int i=0;
    while (i<=3) {
        switch (i) {
            case -3:
            case 3: cout << "s";
                    return 1;
            case 2: cout << "u";
                    break;
            case 0: cout << "m";
                    break;
            default: cout <<"a";
        }
        i=i+1;
    }
    return 0;
}
```

Notieren Sie die Ausgabe

maus

```
#include<iostream.h>
int main() {
    int i=3;
    while (i<=6) {
        switch (i) {
            case 6: cout << "r";
                    break;
            case 2: return 1;
            case 4: cout << "o";
                    break;
            case 3: cout << "r";
                    break;
            default: cout <<"h";
        }
        i=i+1;
    }
    return 0;
}
```

Notieren Sie die Ausgabe

rohr

```
#include<iostream.h>
int main() {
    int i=2,m=1,sum=2;

    while (i<6 && m) {
        sum=sum+i-m;
        i=i+2;
        m=m%2;
    }
    cout << m << i << sum;
    return 0;
}
```

Notieren Sie die Ausgabe

166

```
#include<iostream.h>
int main() {
    int i=4,m=4,sum=2;

    while (m || i<6) {
        sum=sum+m+i;
        m=m-2;
        i=i+1;
    }
    cout << sum << m << i;
    return 0;
}
```

Notieren Sie die Ausgabe

1706

```
#include<iostream.h>
int main() {
    int i=1;

    do{
        cout << "cnt: " << i;
        i=i+2;
    } while (i > 10)

    return 0;
}
```

Notieren Sie die Ausgabe

cnt: 1

```
#include<iostream.h>
int main() {
    int i=10;

    do{
        cout << "cnt: " << i;
        i=i-2;
    } while (i <= 0)

    return 0;
}
```

Notieren Sie die Ausgabe

cnt: 10