

Informatik I

Uebung 3

Gruppe A, ETZ E91

Patrick Boenzli

Aufgabe 1 - Logical Expressions

Vergleichsoperatoren (liefern true/false zurueck)

Ergebnis eines Vergleiches ist immer `true/false`

- `<` kleiner als
- `>` groesser als
- `==` gleich
- `<=` kleiner gleich
- `>=` groesser gleich

Operator: NOT (!)

`!a = c`

a	c
0	1
1	0

Operator: AND (&&)

`a && b = c`

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

Operator: OR (||)

`a || b = c`

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

abnehmende Bindungsprioritaet 

Aufgabe 2 - Bit-Operators

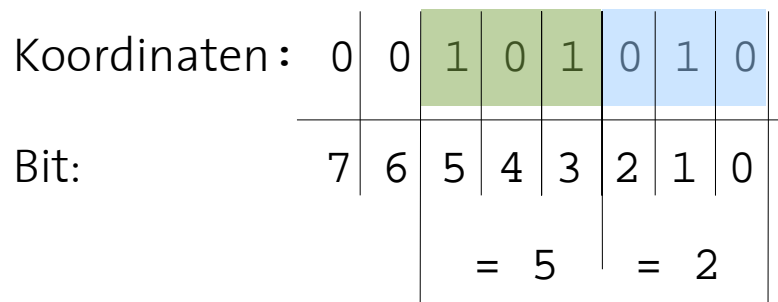
- NOT ~ ~1011 = 0100
- AND & 101 & 110 = 100
- OR | 101 | 110 = 111
- XOR ^ 101 ^ 110 = 011
- L-shift << 101 << 2 = 100
- R-shift >> 101 >> 2 = 001

C++ Primer Plus Seiten 1063 - 1069

Aufgabe 2 - Schach

Adressierung im Schachfeld

- Effiziente codierung der Adresse
- Codiert in 8-Bit Integer Zahl:



	x							
y	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

Aufgabe 2 - Schach

Erreichbarkeit

- Turm: vertikale/horizontale

➔ x und/oder y Komponente
muessen gleich sein

- Beispiel an Wandtafel

y ^x	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

Aufgabe 3 - Branching

if – Anweisung

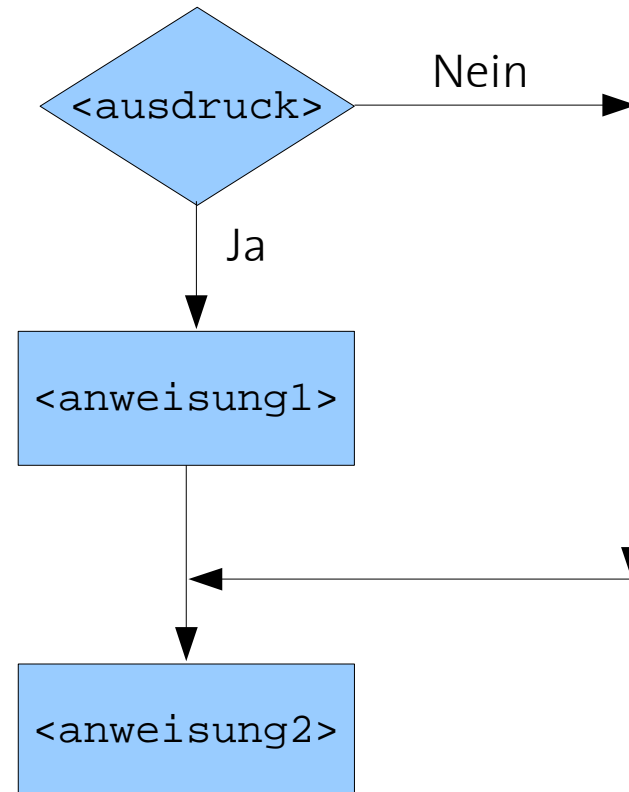
```
if ( <ausdruck> )  
{  
    <anweisung1>  
}
```

<anweisungen2>

- Ist der wert von <ausdruck> ungleich 0 (TRUE) dann wird <anweisung1> ausgefuehrt

- Beispiel auf Computer

C++ Primer Plus Seiten 231 - 236



Aufgabe 3 - Branching

if - Anweisung

```
int a = 4;

if( a == 4)
{
    cout << "out: wir sind im Schleifenblock" << endl;
}

if( a)
    cout << "out: a == true!!" << endl;

a = 0;
if( a)
    cout << "ist jetzt a == true??" << endl;
```

Output:

▼ out: Wir sind im Schleifenblock
▼ out: a == true!!

Aufgabe 4 - Branching

switch - Anweisung

```
switch ( <ausdruck> )
{
    case <ausdruck1>:
        <anweisung1>;
        break;
    case <ausdruck2>:
        <anweisung2>;
        break;
    case <ausdruck3>:
        <anweisung3>;
        break;
    default:
        <default-anweisung>;
}
```

C++ Primer Plus Seiten 251 - 255

Aufgabe 4 - Branching

switch – Anweisung – Einfaches Beispiel

```
int a = 2;

switch( a)
{
    case 0:
        cout << "null" << endl;
        break;
    case 1:
        cout << "eins" << endl;
        break;
    case 2:
        cout << "zwei" << endl;
        break;
    default:
        cout << "unbekannte Zahl" << endl;
}
```

Nachbesprechung Uebung 2

Homepage Uebungsgruppe:

<https://dev.orxonox.net/wiki/boenzlip/inf1>