ETH Zürich

Informatik I
Übungsstunde 02

Johannes Odenthal

cs.jodenthal.com

Hausaufgabenfragen

Welche Fragen zum Tutorial oder zur letzten Übung sind offen?

Integer Division und Modulo

Division und Rest

```
7 / 3
7 % 3
15 / 4
15 % 4
16 / 4
16 % 4
```

`7 / 3 == 2` · `7 % 3 == 1`
`15 / 4 == 3` · `15 % 4 == 3`
`16 / 4 == 4` · `16 % 4 == 0`
`(a / b) * b + a % b == a` für diese Beispiele

Modulo anwenden

```cpp
int a;
std::cin >> a;
if (a % 2 == 0) {
  std::cout << "Yes\n";
} else {
  std::cout << "No\n";
}
```

Welche Frage beantwortet dieses Programm?

Die Ausgabe sagt, ob die eingegebene Zahl gerade ist.

Last Three Digits

CodeExpert

Fragen?

Logical Expressions

Operatorrangfolge

```
1 3 < 4 + 1 && 2 < 3
```

`(3 < (4 + 1)) && (2 < 3)`
`(3 < 5) && true`
`true && true` ⟶ `true`

Short-circuit evaluation · AND

```
1 int x = 1;
2 int y = 0;
3 2 > 3 && 17u - 55 <= ++x % y
```

Die linke Seite ist `false`
Die rechte Seite wird nicht ausgewertet
Ergebnis: `false` · `x` bleibt `1`

Short-circuit evaluation · OR

```
1 int x = 1;
2 x == 1 || 1 / (x - 1) < 1
```

Die linke Seite ist `true`
Die Division wird nicht ausgeführt
Ergebnis: `true`

Von bool zu int

```
1 int x = 1;
2 !(1 < 2 && x == 1) + 1
```

`(!(true && true)) + 1`
`false + 1`
`0 + 1` ⟶ `1`

Boolesche Ausdrücke vereinfachen

```
1 b == true && !(n < 0) && (n != 0)
2     && (n != 6) && (n < 6)
3
4 a && !(a == true) || (b == false)
5
6 !(a != false && !(b == false) == false)
```

a und b: bool · n: int

Welche Teile sind überflüssig?

1: `b && n > 0 && n < 6`
2: `!b`
3: `!a || b`

Fragen?

Mixed Type Expressions

Umwandlungen

```
1 5 / 2
2 5.0 / 2
3 5u + 5 * 3u
```

`5 / 2` ⟶ `2` vom Typ int
`5.0 / 2.0` ⟶ `2.5` vom Typ double
`5u + 15u` ⟶ `20u` vom Typ unsigned int

Ausdrücke auswerten

```
1 (y++ < 0 && y < 0) + 2.0
2
3 y = (x++ = 3)
4
5 3.0 + 3 - 4 + 5
6
7 5 % 4 * 3.0 + true * x++
```

Jeder Ausdruck startet unabhängig mit x = 1 und y = –1

Welche Ausdrücke sind gültig? Bestimme Typ und Wert.

Auswertung · Short-circuit und Post-Increment

```
1 int y = -1;
2 (y++ < 0 && y < 0) + 2.0
```

`(-1 < 0 && y < 0) + 2.0` · jetzt `y == 0`
`(true && false) + 2.0`
`false + 2.0` ⟶ `2.0` · double, rvalue

Zuweisung und Post-Increment

```
1 // Absichtlich ungültig
2 int x = 1;
3 int y;
4 y = (x++ = 3);
```

`x++` liefert den alten Wert als rvalue
Diesem Ergebnis kann kein Wert zugewiesen werden
Der Ausdruck kompiliert nicht

Auswertung · Arithmetik

```
1 3.0 + 3 - 4 + 5
```

`((3.0 + 3.0) - 4) + 5`
`(6.0 - 4.0) + 5`
`2.0 + 5.0` ⟶ `7.0` · double, rvalue

Auswertung · Gemischte Typen

```
1 int x = 1;
2 5 % 4 * 3.0 + true * x++
```

`1 * 3.0 + true * 1` · jetzt `x == 2`
`3.0 + 1 * 1`
`3.0 + 1.0` ⟶ `4.0` · double, rvalue

Fragen?

Optional · Nachschlagen

Gültigkeit und Wertkategorien

```
1 1 * (2 * 3)
2 (a = 1)
3 (1
4 (a * 3) = (b * 5)
```

Was ist gültig?

`1 * (2 * 3)`: gültig, rvalue mit Wert 6
`(a = 1)`: gültig, lvalue; a hat danach Wert 1
Die letzten beiden Zeilen sind absichtlich ungültig