

Exercise Week 01

GianAndrea Müller
`mailto:muellegi@student.ethz`

February 28, 2018

Time Schedule

- 10' Generelle Information
- 15' Ganzzahldivision und Modulo mit Übung
- 10' Repetition einfacher Ausdrücke
- 15' Binäre Darstellung von Zahlen
- 15' Pause
- 40' Übungsbearbeitung

Learning Objectives

- Organisation: Klärung aller Fragen bezüglich der Übungsumgebung.
- Verständnis: Ganzzahldivision
- Verständnis: Ausdrücke
- Verständnis: Binäre Darstellung von Zahlen

Allgemeines

- Selbstverantwortung übernehmen.
- Programmieren ist eine Chance.
- Let me [google](#) that for you.

Organisation

Wieso Übungen lösen?

- Erfahrung sammeln.
- XP sammeln.
- Bonusaufgaben lösen.
- 0.25 Notenpunkte gewinnen.

[Polybox](#)

[First login](#)

[All exercises](#)

[C++ reference](#)

[Hardcore C++ reference](#)

Ein C++ Programm

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main (){
6
7      int a;
8      int b = 4;
9      cin>>a;
10     cout<<a+b<<endl;
11
12     return 0;
13 }
```

Ein C++ Programm

```
1  #include <iostream> //Bibliothek
2
3  using namespace std; //Loesung fuer std::
   cin, std::cin
4
5  int main (){ //Beginn der Hauptfunktion
6
7      int a; //Variabeldefinition ohne
           Initialisierung
8      int b = 4; //Variabeldefinition mit
           Initialisierung
9      cin>>a; //Einlesen mit cin>>
10     cout<<a+b<<endl; //Ausgeben mit cout<<
11     //Neue Zeile mit endl
12     return 0; //Rueckgabewert der
           Hauptfunktion
13 }
```

Ganzzahldivision

```
1 using namespace std;
2
3 int main (){
4     int a = 6;
5     int b = 4;
6     cout<<a/b<<endl;
7     return 0;
8 }
```

Keep in mind!

Eine Ganzzahldivision löscht die Nachkommastellen.

Integer division, examples

```
1  using namespace std;
2
3  int main (){
4      int a = 6;
5      int b = 4;
6      cout<<b/a<<endl;
7      cout<<1/2<<endl;
8      cout<<double(1)/2<<endl;
9      cout<<1.0/2<<endl;
10     return 0;
11 }
```

Ganzzahldivision, Beispiele

```
1  using namespace std;
2
3  int main (){
4      int a = 6;
5      int b = 4;
6      cout<<b/a<<endl;    //0
7      cout<<1/2<<endl;    //0
8      cout<<double(1)/2<<endl;    //0.5
9      cout<<1.0/2<<endl;    //0.5
10     return 0;
11 }
```

Modulo

```
1 using namespace std;
2
3 int main (){
4     int a = 6;
5     int b = 4;
6     cout<<a%b<<endl;
7     return 0;
8 }
```

Keep in mind!

The modulo operation returns the rest of an integer division.

[Short demonstration](#)

Exercise 01_1 ~ 5'

Write a program which reads in two integers a and b , then calculates the quotient $\frac{a}{b}$ as a mixed expression and outputs it.

For example, if $a = 17$ and $b = 6$, the output should be $2 \frac{5}{6}$.

Solution 01_1

[Solution to 01_1](#)

L- und R-Werte

Definition

Ein L-Wert ist ein Ausdruck der über eine Adresse im Computerspeicher verfügt und somit auf der linken Seite eines Zuweisungsoperators (=) stehen kann.

Alle anderen Ausdrücke sind R-Werte.

L- und R-Werte

```
1  using namespace std;
2
3  int main (){
4      int a = 6;
5      int b = 4;
6      //5 = a;
7      //(1+a) = 7;
8      int c = a + b;
9      cout<<a%b<<endl;
10     return 0;
11 }
```

Exercise 01_2 5'

Evaluating expressions

- ❶ Which of the following character sequences are not C++ expressions and why not?
 - ❶ `1*(2*3)`
 - ❷ `(a=1)`
 - ❸ `(1`
 - ❹ `(a*3)=(b*5)`
- ❷ For all of the expressions that you identified in 1), decide whether these are lvalues or rvalues.
- ❸ Determine the values of the expressions and explain how these values are obtained.

[Expressions](#)

[Precedence](#)

Solution 01_2

[Solution 01_2](#)

Binäre Darstellung

$$91310 = 10 * 9131 + 0$$

$$9131 = 10 * 913 + 1$$

$$913 = 10 * 91 + 3$$

$$91 = 10 * 9 + 1$$

$$9 = 10 * 0 + 9$$

$$61 = 2 * 30 + 1$$

$$30 = 2 * 15 + 0$$

$$15 = 2 * 7 + 1$$

$$7 = 2 * 3 + 1$$

$$3 = 2 * 1 + 1$$

$$1 = 2 * 0 + 1$$

Binäre Darstellung

						Σ
Ziffern	9	1	3	1	0	
Multiplikator	10000	1000	100	10	1	
Wert	90000	1000	300	10	0	91310

							Σ
Ziffern	1	1	1	1	0	1	
Multiplikator	32	16	8	4	2	1	
Wert	32	16	8	4	0	1	61

Negative Binärzahlen

bin	uint	int	bin	uint	int
0000	0	0	1000	8	-8
0001	1	1	1001	9	-7
0010	2	2	1010	10	-6
0011	3	3	1011	11	-5
0100	4	4	1100	12	-4
0101	5	5	1101	13	-3
0110	6	6	1110	14	-2
0111	7	7	1111	15	-1

[One's complement](#)

[Demo](#)

[Two's complement](#)

Tips für Exercise 3

- Serienschaltung von R_1 und R_2 : $R_{tot} = R_1 + R_2$
- Parallelschaltung von R_1 und R_2 : $R_{tot} = \frac{R_1 \cdot R_2}{R_1 + R_2}$
- Ganzzahlrundung: $1.999 \rightarrow 1$
- Arithmetische Rundung: $1.5 \rightarrow 2$