Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Programmieren in C

Command Line Arguments (CLA), Operatoren

Dr. Adrian Koller, Büro E307, adrian.koller@hslu.ch

Tagesziele

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Command Line Arguments
- Operatoren
- Workshops (heute neu: «for-loops»)

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Inputs über Kommandozeile

- int main(int argc, char *argv[])
  - argc Anzahl Argumente
  - argv Liste der Argumente
- Übung: Folgendes Programm ausprobieren

```c
#include <stdio.h>

int main( int argc, char *argv[]){

    printf("Number of input arguments: %d\n", argc);

    for(int i = 0; i< argc; i++)
    {
        printf("Argument %d was %s\n", i, argv[i]);
    }

    return 0;
}
```

Inputargumente umwandeln

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Inputargumente sind «strings», das heisst, Text
  - Wir möchten aber «int» haben!
- #include <stdlib.h>

```
int inputValue = atoi(argv[i]);
```

- Achtung: Jeder ungültige Wert (d.h. «nicht-int») wird als 0 interpretiert
- http://www.cplusplus.com/reference/cstdlib/

Übung: Command Line Inputs

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Schreibe ein Programm, welchem genau 3 Command Line Arguments mitgegeben werden müssen.

  Dieses Programm werden wir gleich erweitern, deshalb nennt das Programm

  logicOperators.c

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Zuweisungsoperator =

- variable = ausdruck;
- x = 5;
- y = x;

- gleichzeitige Initialisierung:
  - int x = 5, y = 0, z, a = 2;
  - float e = 2.718281828;

Mathematische Operatoren

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- + Addition
- - Subtraktion
- * Multiplikation
- / Division
- % Modulo

- ++ Inkrement + 1
- -- Dekrement – 1

5%2 ? -> 2

5%2 = 1

10%8 = 2

10%4 = 2

Übungen zu Operatoren

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Schreibe ein Programm um folgende Fälle zu testen:
  - short largeShort = 32760;
    largeShort = largeShort + 12345;
  - int numerator = 19, denominator = 5, result;
    result = numerator/denominator;
  - float e = 2.718281828f;
    e++; // -> print e
    e--;
  - double dblNumerator = 12.8;
    → use % 3

a++ ++a

- Das Programm gibt die Resultate auf der Konsole aus.

Übungen zu Operatoren

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Schreibe folgende Zeilen in ein Programm und beobachte den Output:

```
int a = 0, b = 0;
printf("a: %d\n", ++a);
printf("b: %d\n", b++);

printf("b: %d\n", b);
```

- Was unterscheidet ++a von b++?

Zusammengesetzte Operatoren

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Mathematische Operation mit Zuweisung kombinieren

• x += 5	->	x = x + 5
• x *= y	->	x = x * y
• y -= z + 1	->	y = y – z +1
• a /= b	->	a = a / b
• c %= 3	->	c = c % 3

= Zuweisung

== Vergleich

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Bedingungsoperator '?'

- Einziger Operator mit 3 Operanden
- questionExpression ? resultIfTrue : resultIfFalse
- x = y ? 1 : 100;
  falls y > 0, dann x = 1, sonst x = 100

➔ Übung: Teste die Fälle für y = 200, y = -1, y = 0

- z = (x > y) ? x : y;
  falls x > y, dann z = x, sonst z = y

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Rangfolge von Operatoren

Operator	Priorität
• ++ --	1
• * / %	2
• + -	3

→ Auswertung von links nach rechts bei gleicher Priorität

- Was ist das Resultat von

```
12 % 5 * 2 ?
4 + 5 * 3?
3 * 4 / ++y + 2 / y
```

// anfänglich y = 2

➔ Klammern verwenden!

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Logische Operatoren

&&

- AND && expr1 && expr2
- OR || expr1 || expr2
- NOT ! ! Expr1

& Bitwise
| Bitwise

- Kombinationen:

① ②

- if((expr1 && expr2) || expr3)

true

- if((expr1 || expr2) && !(expr3 && expr4))

else

Logische Operatoren: Übung

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Erweitere das Programm «logicOperators.c»:
  - Beim Start werden 3 Ganzzahlen als CLA mitgegeben.
    → prüfe, dass alle Zahlen > 0 sind, falls nicht, Programm abbrechen
  - Falls Eingabe ok, wird zuerst die kleinste und dann die grösste Zahl ausgegeben.
  - Erwarteter Output:

    The entered values are not all positive. Exiting. // Fehlerfall

    The smallest of the entered values is: 3

    The largest of the entered values is: 12

    The sum of all value is:

    The average of all values: