

1. Tutorium

C - Programme

Rechnerorganisation, Tutorium #13

Patrick Röper | 5. November 2019

FAKULTÄT FÜR INFORMATIK



1 Orga

2 Tools

3 Aufgaben

Name: Patrick Röper

Alter: 21 Jahre

Studiengang: Bachelor Informatik, 5. Semester

Kontakt: roeper.patrick@student.kit.edu

- Min. 10 Übungsblätter.
- 50% der Punkte benötigt.
- **Abgabetermin beachten!**
- Max. 2 Blätter nicht abgeben!
- Regelmäßige Anwesenheit im Tutorium.
- 2 Bonuspunkte für Klausur (ab Note 4.0!)

1 Orga

2 Tools

3 Aufgaben

- Installation der notwendigen Werkzeuge:
`sudo apt-get install build-essential`
- CMake für größere Projekte:
`sudo apt-get install cmake`
- Programm mit beliebigem Texteditor erstellen (gEdit, Kate, Emacs, Vi, ...).
- Programm übersetzen
`gcc -o program program.c`
dabei spezifiziert der Parameter `-o <name>` den Namen des compilierten Programms.
- Programm ausführen
`./program.`

- Installation der notwendigen Werkzeuge: Xcode über mitgelieferte DVD oder über den AppStore beziehen und installieren.
- CMake für größere Projekte: herunterladen von <http://www.cmake.org/> und installieren
- Programm mit beliebigem Texteditor erstellen (Xcode, Emacs, Vi, ...).
- Programm in Terminal übersetzen
gcc -o program program.c dabei spezifiziert der Parameter -o <name> den Namen des compilierten Programms.
- Programm ausführen
./program.

- Visual Studio installieren.
- CMake für größere plattformübergreifende Projekte: herunterladen von <http://www.cmake.org/> und installieren.
- in Visual Studio ein Projekt anlegen.
- Programm in Visual Studio erstellen.
- Programm übersetzen.
- Programm ausführen.

1 Orga

2 Tools

3 Aufgaben

Aufgabe 1.1

Welche Datentypen wurden in der Vorlesung genannt und welche Werte können darin gespeichert werden?

Aufgabe 1.1

Welche Datentypen wurden in der Vorlesung genannt und welche Werte können darin gespeichert werden?

Lösung 1.1

- char: einzelne Zeichen (1 Byte)
- int: Integer-/Ganzzahl (meist 2 od. 4 Byte)
- float: Gleitkommazahl (meist 4 Byte)
- double: Gleitkommazahl doppelter Genauigkeit (meist 8 Byte)

Aufgabe 1.2

Was ist bei diesen Datentypen zu beachten?

Aufgabe 1.2

Was ist bei diesen Datentypen zu beachten?

Lösung 1.2

Die Größe ist abhängig von der Rechnerarchitektur und unterscheidet sich je nach CPU.

Aufgabe 1.3

Wie legt man in C eine Variable von einem entsprechenden Datentyp an und weist dieser einen Wert zu?

Aufgabe 1.3

Wie legt man in C eine Variable von einem entsprechenden Datentyp an und weist dieser einen Wert zu?

Lösung 1.3

```
<Datentyp> <Variablenname>;  
(Bsp: int counter)
```

```
<Datentyp> <Variablenname> = <Initialwert>;  
(Bsp: float ratio = 0.12)
```

Aufgabe 2

Welche Kontrollstrukturen sind in der Programmiersprache C verfügbar und wie werden diese verwendet?

Aufgabe 2

Welche Kontrollstrukturen sind in der Programmiersprache C verfügbar und wie werden diese verwendet?

Lösung 2

```
■ if (Bedingung) {...}
    else {...}

■ switch (<Ausdruck>) {
    case a: Anweisung; //wenn <Ausdruck> = a
    break;
    case b: Anweisung; //wenn <Ausdruck> = b
    break;
    default: Anweisung; break; //sonst
}
```

Lösung 2

```
■ while (<Bedingung >) {  
    ...;  
}  
  
■ do {  
    ...;  
} while (<Bedingung >);  
  
■ for (<Initialisierung >; <Bedingung >;  
      <Aktion nach Schleifendurchlauf >) {  
    ...;  
}
```

Aufgabe 3.1

Wie wird ein Zeiger auf einen Datentyp in C deklariert?

Aufgabe 3.1

Wie wird ein Zeiger auf einen Datentyp in C deklariert?

Lösung 3.1

```
<Datentyp> * <Variablenname>;  
// Bsp: int* pointerToInt;
```

Aufgabe 3.2

Was bedeutet der Ausdruck **&variable** in C?

Aufgabe 3.2

Was bedeutet der Ausdruck **&variable** in C?

Lösung 3.2

&variable erzeugt einen Pointer auf die Variable **variable**. Man erhält damit die Adresse von variable im Speicher.

Aufgabe 3.3

Erklären Sie die folgenden Zeilen C-Code und was am Ende ausgegeben wird (erklären sie dabei, welche Bedeutung die Zeichen & und haben):

```
int a = 12;  
int* p;  
p = &a;  
printf("p = %d, a = %d ", p, *p);
```

Lösung 3.3

```
int a = 12; //Integervariable mit Namen a und Wert 12 anlegen

int* p; //Zeiger auf einen Integer anlegen mit Namen p
        /* wird hier zum Anlegen eines Zeigers verwendet

p = &a; //dem Zeiger p die Adresse der Variablen a zuweisen
        //& wird hier zum Auslesen der Speicheradresse
        //von a verwendet

printf("p = %d, a = %d ", p, *p);
        //zuerst wird der Wert des Zeigers ausgegeben =>
        //Speicheradresse von a
        //diese ist abhaengig von Compiler und
        //Ausfuehrungszeitpunkt dann wird der Inhalt der
        //Speicherstelle ausgegeben, auf die der Zeiger
        //verweist hierzu wird der * verwendet
```


Aufgabe

Erstellen Sie ein Programm in der Programmiersprache C, das zwei Integer-Zahlen addiert und das Ergebnis auf der Kommandozeile ausgibt. Übersetzen Sie das Programm und führen Sie es aus. Erläutern Sie, welche Werkzeuge eingesetzt werden müssen und wie diese installiert werden.

Lösung

```
int main(int argc, char* argv[]) {  
    int number1; int number2; int result;  
  
    number1 = 6;  
    number2 = 7;  
  
    result = number1 + number2;  
  
    printf("%d + %d = %d\n", number1, number2, result);  
  
    return 0;  
}
```