

Auf einen Blick

1	Einstieg in die Welt von C	16
2	Die ersten Schritte	23
3	Grundlegendes zu den Basisdatentypen	37
4	Typenumwandlung (Casts)	76
5	Verzweigungen	88
6	Schleifen – Programmteile wiederholen	114
7	Funktionen erstellen	131
8	Sichtbarkeit, Gültigkeitsbereich und Lebensdauer	150
9	Präprozessor-Direktiven	165
10	Arrays und Zeichenketten (Strings)	183
11	Zeiger (Pointer)	208
12	Dynamische Speicherverwaltung	242
13	Fortgeschrittene Datentypen	262
14	Dynamische Datenstrukturen	295
15	Eingabe- und Ausgabe-Funktionen	309
A	Rangfolge der Operatoren	346
B	Kommandozeilenargumente	350
C	Lösungen der Übungsaufgaben	353

Inhalt

Vorwort	15
1 Einstieg in die Welt von C	16
1.1 Es war einmal	16
1.2 Der ANSI-C-Standard	17
1.3 Was brauche ich für C?	18
1.4 Welches Betriebssystem ...?	20
1.5 Listings zum Buch	20
1.6 Schreibkonventionen im Buch	21
1.7 Aufgaben im Buch	22
1.7.1 Level 1	22
1.7.2 Level 2	22
1.7.3 Level 3	22
2 Die ersten Schritte	23
2.1 Das erste Programm in C	23
2.2 Die Funktion »printf«	25
2.3 Zeichensätze	28
2.3.1 Basis-Zeichensatz	28
2.3.2 Ausführungszeichensatz	28
2.4 Symbole von C	30
2.4.1 Bezeichner	30
2.4.2 Schlüsselwörter	31
2.4.3 Literale	31
2.4.4 Einfache Begrenzer	33
2.5 Kommentare	34
2.6 Aufgaben	35
2.6.1 Level 1	35
2.6.2 Level 2	36
2.6.3 Level 3	36

3	Grundlegendes zu den Basisdatentypen	37
3.1	Deklaration und Definition	37
3.2	Initialisieren einer Variablen	38
3.3	Datentypen für Ganzzahlen	39
3.3.1	Vorzeichenlos und vorzeichenbehaftet	41
3.4	Die Funktion scanf	44
3.5	Fließkommazahlen	45
3.5.1	N-stellige Genauigkeit	47
3.5.2	Nix als Probleme mit Fließkommazahlen	49
3.5.3	Komplexe Gleitkommatypen (C99-Standard)	50
3.6	Rechnen mit Zahlen	52
3.6.1	Arithmetische Operatoren	52
3.6.2	Erweiterte Darstellung arithmetischer Operatoren	55
3.6.3	Mathematische Funktionen	55
3.6.4	Exkurs für Fortgeschrittene: Wertebereich beim Rechnen überschritten	60
3.6.5	Inkrement- und Dekrement-Operator	63
3.6.6	Bit-Operatoren	64
3.6.7	sizeof-Operator	68
3.7	Datentyp für Zeichen	69
3.7.1	Der Datentyp »char«	69
3.7.2	Der Datentyp »wchar_t«	72
3.8	Boolescher Datentyp (C99)	73
3.9	Aufgaben	74
3.9.1	Level 1	74
3.9.2	Level 2	74
3.9.3	Level 3	75

4	Typenumwandlung (Casts)	76
4.1	Implizite Umwandlung des Compilers	76
4.1.1	Die Regeln der impliziten Umwandlung	77
4.2	Explizite Umwandlung durchführen	84
4.3	Aufgaben	85
4.3.1	Level 1	85
4.3.2	Level 2	86
4.3.3	Level 3	87
5	Verzweigungen	88
5.1	Bedingte Anweisung	88
5.1.1	if-Verzweigung	88
5.1.2	Vergleichsoperatoren	90
5.1.3	Alternative else-Verzweigung	92
5.1.4	else-if-Verzweigung	94
5.1.5	Verschachteln von if-Verzweigungen	97
5.2	Der Bedingungsoperator ?:	98
5.3	Logische Verknüpfungen	100
5.3.1	Der !-Operator	101
5.3.2	Der &&-Operator – Logisches UND	102
5.3.3	Der -Operator – Logisches ODER	103
5.4	Die Fallunterscheidung – switch	105
5.5	Aufgaben	110
5.5.1	Level 1	110
5.5.2	Level 2	111
5.5.3	Level 3	112
6	Schleifen – Programmteile wiederholen	114
6.1	Die Zählschleife – for	114
6.2	Die kopfgesteuerte while-Schleife	118
6.3	Die fußgesteuerte do-while-Schleife	120

6.4	Kontrollierte Sprünge aus Schleifen	122
6.5	Endlosschleifen	125
6.6	Fehlervermeidung bei Schleifen	126
6.7	Aufgaben	128
6.7.1	Level 1	129
6.7.2	Level 2	129
6.7.3	Level 3	130
7	Funktionen erstellen	131
7.1	Funktionen definieren	131
7.2	Funktionen aufrufen	132
7.3	Funktionsdeklaration (Vorwärts-Deklaration)	133
7.4	Funktionsparameter (call-by-value)	134
7.5	Rückgabewert von Funktionen	136
7.6	Exkurs: Funktion bei der Ausführung	139
7.7	Inline-Funktionen (C99)	140
7.8	Rekursionen	142
7.9	main-Funktion	143
7.10	Aufgaben	146
7.10.1	Level 1	146
7.10.2	Level 2	147
7.10.3	Level 3	147
8	Sichtbarkeit, Gültigkeitsbereich und Lebensdauer	150
8.1	Lokale und globale Variablen	150
8.1.1	Lokale Variablen	150
8.1.2	Globale Variablen	151
8.2	Gültigkeitsbereich	154
8.3	Lebensdauer	155
8.4	Speicherklassen-Spezifizierer	156
8.4.1	Das Schlüsselwort »auto«	156
8.4.2	Das Schlüsselwort »extern«	157

8.4.3	Das Schlüsselwort »static«	158
8.4.4	Das Schlüsselwort »register«	160
8.5	Typ-Qualifizierer	160
8.6	Aufgaben	163
8.6.1	Level 1	163
8.6.2	Level 2	163
9	Präprozessor-Direktiven	165
9.1	Dateien einfügen mit »#include«	165
9.2	Konstanten und Makros mit »#define« und »#undef«	167
9.2.1	Symbolische Konstanten mit »#define«	167
9.2.2	Makros mit »#define«	170
9.2.3	Symbolische Konstanten und Makros aufheben (#undef)	172
9.3	Bedingte Kompilierung	173
9.4	Weitere Präprozessor-Direktiven	176
9.5	Aufgaben	179
9.5.1	Level 1	180
9.5.2	Level 2	180
9.5.3	Level 3	181
10	Arrays und Zeichenketten (Strings)	183
10.1	Arrays verwenden	183
10.1.1	Arrays definieren	183
10.1.2	Arrays initialisieren und darauf zugreifen	184
10.1.3	Arrays mit »scanf« einlesen	191
10.1.4	Arrays an Funktionen übergeben	192
10.2	Mehrdimensionale Arrays	193
10.2.1	Zweidimensionale Arrays initialisieren und darauf zugreifen	194

10.2.2	Zweidimensionale Arrays an eine Funktion übergeben	197
10.2.3	Noch mehr Dimensionen	199
10.3	Strings (Zeichenketten)	199
10.3.1	Strings initialisieren	200
10.3.2	Einlesen von Strings	202
10.3.3	Stringfunktionen der Standard-Bibliothek – <string.h>	203
10.3.4	Umwandlungsfunktionen zwischen Zahlen und Strings	205
10.4	Aufgaben	205
10.4.1	Level 1	205
10.4.2	Level 2	206
10.4.3	Level 3	207
11	Zeiger (Pointer)	208
11.1	Zeiger deklarieren	208
11.2	Zeiger initialisieren	209
11.3	Zugriff auf dem Inhalt von Zeigern	212
11.4	Speichergröße von Zeigern	216
11.5	Zeiger-Arithmetik	216
11.6	Zeiger als Funktionsparameter (call-by-reference)	217
11.7	Zeiger als Rückgabewert	218
11.8	Arrays bzw. Strings und Zeiger	221
11.8.1	Zugriff auf Array-Elemente über Zeiger	221
11.8.2	Array und Zeiger als Funktionsparameter	223
11.8.3	char-Arrays und Zeiger	224
11.9	Zeiger-Arrays	225
11.10	Zeiger auf Arrays	228
11.11	Zeiger auf Zeiger (Zeigerzeiger)	229
11.12	void-Zeiger	230

11.13	Typ-Qualifizierer bei Zeigern	232
11.13.1	Konstanter Zeiger	232
11.13.2	Readonly-Zeiger (Konstante Daten)	232
11.13.3	Konstante Parameter für Funktionen	233
11.13.4	restrict-Zeiger (C99)	233
11.14	Zeiger auf Funktionen	235
11.15	Aufgaben	238
11.15.1	Level 1	238
11.15.2	Level 2	239
11.15.3	Level 3	241
12	Dynamische Speicherverwaltung	242
12.1	Neuen Speicherblock reservieren	243
12.2	Speicherblock vergrößern oder verkleinern	247
12.3	Speicherblock freigeben	250
12.4	Die Heap-Fragmentierung	254
12.5	Zweidimensionale dynamische Arrays	257
12.6	Aufgaben	259
12.6.1	Level 1	259
12.6.2	Level 2	259
12.6.3	Level 3	261
13	Fortgeschrittene Datentypen	262
13.1	Strukturen	262
13.1.1	Struktur deklarieren	263
13.1.2	Definition einer Strukturvariablen	264
13.1.3	Deklaration und Definition zusammen- fassen	264
13.1.4	Synonyme für Strukturtypen erstellen	265
13.1.5	Zugriff auf Strukturelemente	266
13.1.6	Strukturen initialisieren	268
13.1.7	Nur bestimmte Elemente initialisieren (C99)	269

13.1.8	Operationen auf Strukturen	270
13.1.9	Strukturen, Funktionen und Struktur- zeiger	270
13.1.10	Arrays von Strukturen	273
13.1.11	Strukturen in Strukturen	275
13.1.12	Zeiger in Strukturen	279
13.2	Union	281
13.3	Bitfelder	285
13.4	Das offsetof-Makro	289
13.5	Der Aufzählungstyp »enum«	289
13.6	Eigene Typen mit »typedef«	291
13.7	Aufgaben	292
13.7.1	Level 1	292
13.7.2	Level 2	292
13.7.3	Level 3	293
14	Dynamische Datenstrukturen	295
14.1	Verkettete Liste	295
14.1.1	Neues Element in der Liste einfügen	300
14.1.2	Element ausgeben (und suchen)	302
14.1.3	Element aus der Liste entfernen	303
14.2	Doppelt verkettete Listen	306
14.3	Aufgaben	307
14.3.1	Level 1	307
14.3.2	Level 2	307
14.3.3	Level 3	308
15	Eingabe- und Ausgabe-Funktionen	309
15.1	Verschiedene Streams und Standard-Streams	309
15.1.1	Text-Streams	309
15.1.2	Binäre Streams	310
15.1.3	Standard-Streams	310

15.2	Dateien	311
15.3	Dateien öffnen	311
15.4	Dateien schließen	315
15.5	Lesen und Schreiben	316
15.6	Funktionen zur unformatierten Ein-/Ausgabe	316
15.6.1	Einzelne Zeichen lesen	316
15.6.2	Einzelne Zeichen schreiben	317
15.6.3	Zeilenweise (bzw. String) einlesen	319
15.6.4	Zeilenweise (bzw. String) schreiben	320
15.6.5	Blockweise lesen und schreiben	323
15.7	Funktionen zur formatierten Ein-/Ausgabe	326
15.7.1	Funktionen zur formatierten Ausgabe	327
15.7.2	Funktionen zur formatierten Eingabe	333
15.8	Wahlfreier Dateizugriff	335
15.8.1	Dateiposition ermitteln	336
15.8.2	Dateiposition ändern	336
15.9	Fehlerbehandlung	339
15.9.1	Fehler-Flag von Stream überprüfen – »ferror()«	339
15.9.2	Dateiende von Stream überprüfen – »feof()«	340
15.9.3	Die Fehlervariable »errno«	341
15.9.4	Fehler- und EOF-Flag zurücksetzen – »clearerr()«	342
15.10	Datei löschen oder umbenennen	342
15.11	Pufferung	342
15.12	Aufgabe	343
15.12.1	Level 1	343
15.12.2	Level 2	343
15.12.3	Level 3	345

A	Rangfolge der Operatoren	346
A.1	Operatoren-Priorität	346
A.2	ASCII-Code-Tabelle	348
A.3	Reservierte Schlüsselwörter in C	349
A.4	Standard-Headerdateien der ANSI-C-Bibliothek	349
B	Kommandozeilenargumente	350
C	Lösungen der Übungsaufgaben	353
	Index	398