

## Auf einen Blick

1	Einstieg in die Welt von C .....	21
2	Erste Schritte in C .....	37
3	Basisdatentypen in C .....	55
4	Rechnen mit C und Operatoren .....	85
5	Bedingte Anweisung und Verzweigung .....	115
6	Schleifen – Programmteile wiederholen .....	147
7	Funktionen erstellen .....	163
8	Präprozessor-Direktiven .....	193
9	Arrays und Zeichenketten (Strings) .....	219
10	Zeiger (Pointer) .....	251
11	Dynamische Speicherverwaltung .....	289
12	Komplexe Datentypen .....	305
13	Dynamische Datenstrukturen .....	343
14	Eingabe- und Ausgabefunktionen .....	361
15	Zeitroutinen (time.h) .....	411

# Inhalt

Vorwort .....	17
<b>1 Einstieg in die Welt von C</b>	<b>21</b>
<hr/>	
1.1 Die Sprache C .....	21
1.2 Die C-Standardbibliothek .....	22
1.2.1 Header-Dateien und Programmbibliothek .....	23
1.3 Die nötigen Werkzeuge für C .....	26
1.4 Übersetzen mit der Entwicklungsumgebung .....	29
1.5 Übersetzen mit GCC und Clang .....	33
1.6 Listings zum Buch .....	35
1.7 Kontrollfragen und Aufgaben im Buch .....	36
<b>2 Erste Schritte in C</b>	<b>37</b>
<hr/>	
2.1 Das erste Programm in C .....	37
2.2 Die Funktion printf() .....	40
2.3 Zeichensätze in C .....	42
2.3.1 Basis-Ausführungszeichensatz .....	42
2.3.2 Einige wichtige Escape-Sequenzen .....	42
2.4 Symbole in C .....	45
2.4.1 Bezeichner .....	45
2.4.2 Reservierte Schlüsselwörter .....	46
2.4.3 Literale .....	48
2.4.4 Einfache Begrenzer .....	49

<b>2.5</b>	<b>Kommentare</b> .....	50
<b>2.6</b>	<b>Kontrollfragen und Aufgaben</b> .....	52
<b>3</b>	<b>Basisdatentypen in C</b> .....	55
<b>3.1</b>	<b>Variablen</b> .....	55
<b>3.2</b>	<b>Deklaration und Definition</b> .....	56
<b>3.3</b>	<b>Initialisierung und Zuweisung von Werten</b> .....	58
<b>3.4</b>	<b>Datentypen für Ganzzahlen</b> .....	59
3.4.1	Vorzeichenlos und vorzeichenbehaftet .....	61
3.4.2	Suffixe für Ganzzahlen .....	64
<b>3.5</b>	<b>Datentypen für Zeichen</b> .....	65
3.5.1	Der Datentyp char .....	65
3.5.2	Der Datentyp wchar_t .....	67
3.5.3	Unicode-Unterstützung .....	68
<b>3.6</b>	<b>Datentypen für Fließkommazahlen</b> .....	68
3.6.1	Suffixe für Fließkommazahlen .....	70
3.6.2	Komplexe Gleitkommatypen .....	70
<b>3.7</b>	<b>Boolescher Datentyp</b> .....	72
<b>3.8</b>	<b>Speicherbedarf mit sizeof ermitteln</b> .....	73
<b>3.9</b>	<b>Wertebereiche der Datentypen ermitteln</b> .....	75
3.9.1	Limits von Integer-Typen .....	76
3.9.2	Limits von Fließkommazahlen .....	78
3.9.3	Integer-Typen mit fester Größe verwenden .....	79
3.9.4	Sicherheit beim Kompilieren mit <code>_static_assert()</code> ...	80
<b>3.10</b>	<b>Konstanten erstellen</b> .....	81
<b>3.11</b>	<b>Lebensdauer und Sichtbarkeit von Variablen</b> .....	82

<b>3.12</b>	<b>void – ein unvollständiger Typ</b> .....	84
<b>3.13</b>	<b>Kontrollfragen und Aufgaben</b> .....	84
<b>4</b>	<b>Rechnen mit C und Operatoren</b> .....	85
<b>4.1</b>	<b>Werte formatiert einlesen mit scanf()</b> .....	85
<b>4.2</b>	<b>Operatoren</b> .....	89
<b>4.3</b>	<b>Arithmetische Operatoren</b> .....	93
<b>4.4</b>	<b>Inkrement- und Dekrementoperator</b> .....	95
<b>4.5</b>	<b>Bitoperatoren</b> .....	98
<b>4.6</b>	<b>Implizite Typumwandlung</b> .....	102
4.6.1	Arithmetische Umwandlung .....	103
4.6.2	Typpromotionen .....	105
<b>4.7</b>	<b>Explizite Typumwandlung</b> .....	106
<b>4.8</b>	<b>Mathematische Funktionen in C</b> .....	107
<b>4.9</b>	<b>Kontrollfragen und Aufgaben</b> .....	112
<b>5</b>	<b>Bedingte Anweisung und Verzweigung</b> .....	115
<b>5.1</b>	<b>Bedingte Anweisungen</b> .....	115
5.1.1	Bedingte Anweisungen mit if .....	116
5.1.2	Vergleichsoperatoren .....	118
5.1.3	Mini-Exkurs: Anweisungsblock .....	121
<b>5.2</b>	<b>Die alternative Verzweigung</b> .....	121
<b>5.3</b>	<b>Der Bedingungsoperator ?:</b> .....	124
<b>5.4</b>	<b>Die mehrfache Verzweigung mit if und else if</b> .....	125
5.4.1	Verschachteln von Verzweigungen .....	128

<b>5.5 Mehrfache Verzweigung mit switch</b> .....	130
5.5.1 Austritt aus der Fallunterscheidung mit break .....	133
<b>5.6 Logische Verknüpfungen</b> .....	136
5.6.1 Der !-Operator .....	137
5.6.2 Der &&-Operator – logisches UND .....	139
5.6.3 Der   -Operator – logisches ODER .....	141
<b>5.7 Kontrollfragen und Aufgaben</b> .....	143
<b>6 Schleifen – Programmteile wiederholen</b> .....	147
<hr/>	
<b>6.1 Die Zählschleife – for</b> .....	147
<b>6.2 Die kopfgesteuerte while-Schleife</b> .....	152
<b>6.3 Die fußgesteuerte do-while-Schleife</b> .....	154
<b>6.4 Kontrolliertes Verlassen von Schleifen</b> .....	157
6.4.1 break .....	157
6.4.2 continue .....	159
<b>6.5 Kontrollfragen und Aufgaben</b> .....	161
<b>7 Funktionen erstellen</b> .....	163
<hr/>	
<b>7.1 Funktionen definieren</b> .....	164
<b>7.2 Funktionen aufrufen</b> .....	165
<b>7.3 Funktionsdeklaration (Vorausdeklaration)</b> .....	166
<b>7.4 Funktionsparameter</b> .....	169
<b>7.5 Rückgabewert von Funktionen</b> .....	171
7.5.1 Die return-Anweisung .....	173
<b>7.6 Exkurs: Funktionen bei der Ausführung</b> .....	175

<b>7.7 Inline-Funktionen</b> .....	176
<b>7.8 Rekursionen</b> .....	177
<b>7.9 Die main()-Funktion</b> .....	179
7.9.1 Der Rückgabewert von main() an das Betriebssystem .....	180
<b>7.10 Programme mit exit() beenden</b> .....	182
<b>7.11 Globale, lokale und statische Variablen</b> .....	183
7.11.1 Lokale Variablen .....	184
7.11.2 Globale Variablen .....	185
7.11.3 Speicherklasse static .....	188
7.11.4 Die Speicherklasse extern .....	190
<b>7.12 Kontrollfragen und Aufgaben</b> .....	191
<b>8 Präprozessor-Direktiven</b> .....	193
<hr/>	
<b>8.1 Dateien einfügen mit #include</b> .....	194
<b>8.2 Konstanten und Makros mit #define und #undef</b> .....	195
8.2.1 Symbolische Konstanten mit #define definieren ...	196
8.2.2 Makros mit #define definieren .....	198
8.2.3 Symbolische Konstanten und Makros aufheben (#undef) .....	202
<b>8.3 Bedingte Kompilierung</b> .....	202
8.3.1 Mehrfaches Inkludieren vermeiden .....	205
<b>8.4 Programmdiagnose mit assert()</b> .....	209
<b>8.5 Generische Auswahl</b> .....	211
<b>8.6 Eigene Header erstellen</b> .....	213
<b>8.7 Kontrollfragen und Aufgaben</b> .....	215

<b>9</b>	<b>Arrays und Zeichenketten (Strings)</b>	219
<b>9.1</b>	<b>Was genau sind Arrays?</b>	219
9.1.1	Arrays definieren	220
9.1.2	Arrays mit Werten versehen und darauf zugreifen	221
9.1.3	Initialisierung mit einer Initialisierungsliste	225
9.1.4	Bestimmte Elemente direkt initialisieren	227
9.1.5	Array mit Schreibschutz	227
9.1.6	Arrays mit fester und variabler Länge (VLA) – optional seit C11	227
9.1.7	Arrays mit scanf einlesen	229
9.1.8	Arrays an Funktionen übergeben	230
<b>9.2</b>	<b>Mehrdimensionale Arrays</b>	232
9.2.1	Zweidimensionalen Arrays Werte zuweisen und darauf zugreifen	233
9.2.2	Zweidimensionale Arrays an eine Funktion übergeben	236
9.2.3	Können wir noch mehr Dimensionen verwenden?	238
<b>9.3</b>	<b>Strings (Zeichenketten)</b>	239
9.3.1	Strings initialisieren	239
9.3.2	Einlesen von Strings	241
9.3.3	Unicode-Unterstützung	243
9.3.4	String-Funktionen der Standardbibliothek string.h	245
9.3.5	Sicherere Funktionen zum Schutz vor Speicherüberschreitungen	248
9.3.6	Umwandlungsfunktionen zwischen Zahlen und Strings	248
<b>9.4</b>	<b>Kontrollfragen und Aufgaben</b>	249

<b>10</b>	<b>Zeiger (Pointer)</b>	251
<b>10.1</b>	<b>Zeiger vereinbaren</b>	251
<b>10.2</b>	<b>Zeiger verwenden</b>	252
10.2.1	Explizite Typumwandlung für den byteweisen Zugriff	254
<b>10.3</b>	<b>Zugriff auf den Inhalt von Zeigern</b>	255
10.3.1	Der NULL-Zeiger	257
10.3.2	Deklaration, Adressierung und Dereferenzierung von Zeigern	259
<b>10.4</b>	<b>Zeiger als Funktionsparameter</b>	260
<b>10.5</b>	<b>Zeiger als Rückgabewert</b>	261
<b>10.6</b>	<b>Zeigerarithmetik</b>	264
<b>10.7</b>	<b>Zugriff auf Array-Elemente über Zeiger</b>	265
<b>10.8</b>	<b>Arrays und Zeiger als Funktionsparameter</b>	268
10.8.1	const-Array-Parameter	269
<b>10.9</b>	<b>char-Arrays und Zeiger</b>	271
<b>10.10</b>	<b>Arrays von Zeigern</b>	272
<b>10.11</b>	<b>void-Zeiger</b>	275
<b>10.12</b>	<b>Typqualifizierer bei Zeigern</b>	276
10.12.1	Konstanter Zeiger	277
10.12.2	Zeiger für konstante Daten	277
10.12.3	Konstante Zeiger und Zeiger für konstante Daten	278
10.12.4	Konstante Parameter für Funktionen	278
10.12.5	restrict-Zeiger	279
<b>10.13</b>	<b>Zeiger auf Funktionen</b>	281
<b>10.14</b>	<b>Kontrollfragen und Aufgaben</b>	285

<b>11 Dynamische Speicherverwaltung</b>	289
<b>11.1 Neuen Speicher zur Laufzeit reservieren</b>	290
<b>11.2 Speicherblöcke vergrößern oder verkleinern</b>	295
<b>11.3 Speicherblöcke wieder freigeben</b>	300
11.3.1 Memory Leaks (Speicherlecks)	300
<b>11.4 Kontrollfragen und Aufgaben</b>	302
<b>12 Komplexe Datentypen</b>	305
<b>12.1 Strukturen</b>	306
12.1.1 Strukturtypen deklarieren	306
12.1.2 Definition einer Strukturvariablen	308
12.1.3 Erlaubte Operationen auf Strukturvariablen	308
12.1.4 Deklaration und Definition zusammenfassen	309
12.1.5 Synonyme für Strukturtypen erstellen	310
12.1.6 Selektion von Komponenten einer Strukturvariablen	310
12.1.7 Strukturen initialisieren	314
12.1.8 Nur bestimmte Komponenten einer Strukturvariablen initialisieren	315
12.1.9 Zuweisung bei Strukturvariablen	316
12.1.10 Größe und Speicherausrichtung einer Struktur	317
12.1.11 Strukturen vergleichen	317
12.1.12 Strukturen, Funktionen und Strukturzeiger	317
12.1.13 Array von Strukturvariablen	322
12.1.14 Strukturvariablen als Komponenten in Strukturen	325
12.1.15 Zeiger als Komponenten in Strukturen	330
<b>12.2 Unions</b>	333

<b>12.3 Der Aufzählungstyp enum</b>	336
<b>12.4 Eigene Typen mit typedef deklarieren</b>	338
<b>12.5 Kontrollfragen und Aufgaben</b>	340
<b>13 Dynamische Datenstrukturen</b>	343
<b>13.1 (Einfach) verkettete Listen</b>	343
13.1.1 Ein neues Element in die Liste einfügen	350
13.1.2 Ein Element suchen und ausgeben	352
13.1.3 Ein Element aus der Liste entfernen	353
13.1.4 Das erste Element in der Liste löschen	354
13.1.5 Ein beliebiges Element in der Liste löschen	355
<b>13.2 Doppelt verkettete Listen</b>	357
<b>13.3 Kontrollfragen und Aufgaben</b>	358
<b>14 Eingabe- und Ausgabefunktionen</b>	361
<b>14.1 Verschiedene Streams und Standard-Streams</b>	361
14.1.1 Streams im Textmodus	362
14.1.2 Streams im binären Modus	363
14.1.3 Standard-Streams	363
<b>14.2 Dateien</b>	364
<b>14.3 Dateien öffnen</b>	365
14.3.1 Exklusiver Dateizugriff	368
14.3.2 Weitere Dateifunktionen	369
<b>14.4 Dateien schließen</b>	371
14.4.1 Limit maximal geöffneter Dateien	371

<b>14.5 Fehler oder Dateiende prüfen</b> .....	372
14.5.1 End-of-File Indicator .....	372
14.5.2 Error Indicator .....	373
14.5.3 Informationen zum Fehler ausgeben .....	373
14.5.4 Fehler- und EOF-Flag zurücksetzen – clearerr() .....	374
<b>14.6 Weitere Funktionen für die Ein- und Ausgabe</b> .....	374
14.6.1 Einzelne Zeichen aus einem Stream lesen .....	375
14.6.2 Zeichen in den Stream zurückstellen .....	375
14.6.3 Einzelne Zeichen in einen Stream schreiben .....	375
14.6.4 Zeilenweise aus einem Stream lesen .....	378
14.6.5 Zeilenweise in einen Stream schreiben .....	378
14.6.6 Lesen und Schreiben in ganzen Blöcken .....	383
<b>14.7 Funktionen zur formatierten Ein-/Ausgabe</b> .....	387
14.7.1 Funktionen zur formatierten Ausgabe .....	388
14.7.2 Umwandlungsvorgaben für die printf-Familie .....	389
14.7.3 Weite und Feldbreite .....	390
14.7.4 Flags .....	391
14.7.5 Genauigkeitsangaben für Zahlen bei printf() .....	393
14.7.6 Funktionen für die formatierte Eingabe .....	396
14.7.7 Umwandlungsvorgaben für die scanf()-Familie .....	397
14.7.8 Suchmengenkonvertierung .....	398
<b>14.8 Wahlfreier Dateizugriff</b> .....	400
14.8.1 Die aktuelle Dateiposition ermitteln .....	400
14.8.2 Die aktuelle Dateiposition ändern .....	400
<b>14.9 Sicherere Funktionen mit C11</b> .....	404
<b>14.10 Datei löschen oder umbenennen</b> .....	405
<b>14.11 Pufferung</b> .....	405
<b>14.12 Kontrollfragen und Aufgaben</b> .....	407

<b>15 Zeitroutinen (time.h)</b> .....	411
<b>15.1 Die Funktion clock()</b> .....	411
<b>15.2 Erweiterte Zeitfunktionen</b> .....	413
15.2.1 Eine Plattformunabhängige delay()-Funktion .....	415
15.2.2 Der Datentyp time_t .....	416
15.2.3 Der Datentyp struct tm .....	418
<b>Anhang</b> .....	427
<b>A Übersichtstabellen wichtiger Sprachelemente</b> .....	427
A.1 Operatorpriorität (Operator Precedence) .....	427
A.2 Reservierte Schlüsselwörter in C .....	429
A.3 Header-Dateien der Standardbibliothek .....	430
A.4 Kommandozeilenargumente .....	432
A.5 Weiterführende Ressourcen .....	435
<b>B Lösungen der Übungsaufgaben</b> .....	437
B.1 Antworten und Lösungen zu Kapitel 2 .....	437
B.2 Antworten und Lösungen zu Kapitel 3 .....	438
B.3 Antworten und Lösungen zu Kapitel 4 .....	438
B.4 Antworten und Lösungen zu Kapitel 5 .....	440
B.5 Antworten und Lösungen zu Kapitel 6 .....	443
B.6 Antworten und Lösungen zu Kapitel 7 .....	446
B.7 Antworten und Lösungen zu Kapitel 8 .....	448
B.8 Antworten und Lösungen zu Kapitel 9 .....	451
B.9 Antworten und Lösungen zu Kapitel 10 .....	456
B.10 Antworten und Lösungen zu Kapitel 11 .....	461
B.11 Antworten und Lösungen zu Kapitel 12 .....	465