

Inhaltsverzeichnis

1	Einführung in die Programmiersprache C	1
1.1	Hello World	2
1.2	Die Entwicklung von C	6
1.3	Abstraktion	12
1.4	Vorgänger und Nachfolger von C	16
1.5	Standardisierung von C	18
2	Einfache Beispielprogramme	21
2.1	Quadratzahlen	21
2.2	Celsius und Fahrenheit	28
2.3	Zinsberechnung	34
3	Entwurf von Programmen	39
3.1	Vom Problem zum Programm	39
3.2	Entwurf mit Flussdiagrammen nach UML	46
3.3	Pseudocode	54
3.4	Das finale Programm von Euklid in C	56
3.5	Übungsaufgaben	57
4	Aufbau eines Programmes in C	59
4.1	Zeichen und Codierungen	59
4.2	Variablen und Werte	64
4.3	Funktionen in prozeduralen Programmen	68
4.4	Schrittweise Verfeinerung beim Top-Down-Design	73
4.5	Funktionen in C	77
4.6	Struktur einer Quelldatei in C	79
5	Programmerzeugung und -ausführung	85
5.1	Compiler	87
5.2	Linker	93
5.3	Lader	95
5.4	Integrierte Entwicklungsumgebungen und Debugger	97
6	Lexikalische Konventionen	99
6.1	Zeichenvorrat von C	99
6.2	Lexikalische Einheiten	101
6.3	Reservierte Wörter (Schlüsselwörter)	103
6.4	Bezeichner	105
6.5	Literale Konstanten	107
6.6	Operatoren und Interpunktionszeichen	115
6.7	Übungsaufgaben	117

7	Datentypen und Variablen in C	119
7.1	Typkategorien	120
7.2	Elementare Datentypen in C	121
7.3	Modifikatoren und andere Besonderheiten	132
7.4	Variablen in C	135
7.5	Qualifikatoren	141
7.6	Übungsaufgaben	144
8	Einführung in Pointer und Arrays	145
8.1	Pointer-Typen und Pointer-Variablen	145
8.2	Pointer auf void	158
8.3	Arrays	159
8.4	Der Qualifikator restrict	166
8.5	Übungsaufgaben	168
9	Ausdrücke, Anweisungen und Operatoren	171
9.1	Ausdrücke und Anweisungen	171
9.2	Operatoren und Operanden	180
9.3	Einstellige arithmetische Operatoren	187
9.4	Zweistellige arithmetische Operatoren	189
9.5	Zuweisungs-Operatoren	192
9.6	Relationale Operatoren	194
9.7	Logische Operatoren	197
9.8	Bit-Operatoren	201
9.9	Sonstige Operatoren	206
9.10	Implizite Typumwandlung	217
9.11	Konvertierungsvorschriften	220
9.12	Übungsaufgaben	223
10	Kontrollstrukturen	229
10.1	Blöcke – Kontrollstruktur für die Sequenz	229
10.2	if und else – Einfache Selektion	230
10.3	switch – Fallunterscheidung	234
10.4	while – Bedingte Schleife	241
10.5	for – Zählschleife	242
10.6	do while – Annehmende bedingte Schleife	249
10.7	Endlosschleifen und leere Ausdrücke bei Schleifen	250
10.8	break – Abbruch-Anweisung	253
10.9	continue – Fortsetzungs-Anweisung	254
10.10	return – Rücksprung-Anweisung	255
10.11	goto – Sprunganweisung und Marken	255
10.12	Übungsaufgaben	257

11	Blöcke und Funktionen	259
11.1	Struktur und Schachtelung von Blöcken	260
11.2	Lebensdauer, Gültigkeit und Sichtbarkeit von Variablen	262
11.3	Definition von Funktionen	267
11.4	Rücksprung aus einer Funktion – die return-Anweisung	273
11.5	Konventionen bei der Parameterübergabe	275
11.6	Vorwärtsdeklaration von Funktionen	280
11.7	Die Ellipse . . . – variable Parameteranzahl	284
11.8	Iteration und Rekursion	287
11.9	inline-Funktionen	298
11.10	Funktionen ohne Wiederkehr in C11	300
11.11	Übungsaufgaben	301
12	Fortgeschrittene Programmierung mit Pointern	305
12.1	Gleichheit und Unterschiede von Arrays und Pointer	305
12.2	Pointerarithmetik	308
12.3	Initialisierung von Arrays	312
12.4	Übergabe von Arrays an Funktionen	319
12.5	Unterschiede zwischen char-Arrays und Pointern auf char	322
12.6	Das Schlüsselwort const bei Pointern und Arrays	324
12.7	Beispiel für Strings und Pointerarithmetik	326
12.8	Funktionen zur Speicherbearbeitung	329
12.9	Standardfunktionen zur Stringverarbeitung	335
12.10	Weitere Funktionen in <string.h>	343
12.11	Beispiele für Arrays und Pointer	344
12.12	Pointer auf Funktionen	350
12.13	Pointer auf void für Strukturen	354
12.14	Übungsaufgaben	357
13	Strukturen, Unionen und Bitfelder	361
13.1	Strukturen	362
13.2	Anonyme Strukturen	381
13.3	Unionen	384
13.4	Bitfelder	388
13.5	Übungsaufgaben	391
14	Komplexere Vereinbarungen, eigene Typnamen und Namensräume	393
14.1	Komplexere Vereinbarungen	393
14.2	Vereinbarung eigener Typnamen mittels typedef	396
14.3	Namensräume	401
15	Speicherklassen	403
15.1	Der Linker	405

15.2	Der Lader und die Segmente des Adressraums eines Programms	408
15.3	Die Speicherklasse extern bei Programmen aus mehreren Dateien	411
15.4	Die Speicherklasse static bei Programmen aus mehreren Dateien	415
15.5	Die Speicherklasse auto bei lokalen Variablen	417
15.6	Die Speicherklasse register bei lokalen Variablen	418
15.7	Die Speicherklasse static bei lokalen Variablen	420
15.8	Automatische und nicht automatische Initialisierung	422
15.9	Überblick über die Speicherklassen	423
15.10	Übungsaufgaben	424
16	Ein- und Ausgabe	427
16.1	Speicherung von Daten in Dateisystemen	428
16.2	Das Ein-/Ausgabe-Konzept von C	432
16.3	Standardeingabe und Standardausgabe in C	434
16.4	High- und Low-Level-Bibliotheksfunktionen zur Ein- und Ausgabe	438
16.5	Der File-Pointer bei High-Level-Funktionen	440
16.6	Einfaches Schreiben in Dateien mit High-Level-Funktionen	450
16.7	Schreiben von formatierten Strings	453
16.8	Lesen von Dateien mit High-Level-Funktionen	461
16.9	Lesen von formatierten Strings	465
16.10	Alternative Funktionen für mehr Sicherheit nach C11	470
16.11	Übungsaufgaben	474
17	Programmaufruf und -beendigung	477
17.1	Übergabe von Argumenten beim Programmaufruf	477
17.2	Beendigung von Programmen	481
18	Dynamische Speicherzuweisung	489
18.1	Reservierung von Speicher	491
18.2	Freigabe von Speicher mittels free()	497
18.3	Dynamisch erzeugte Arrays	500
18.4	Übungsaufgaben	502
19	Dynamische Datenstrukturen	505
19.1	Verkettete Listen	505
19.2	Baumstrukturen	515
19.3	Übungsaufgaben	528
20	Sortieren und Suchen	531
20.1	Interne Sortierverfahren	532
20.2	Einfache Suchverfahren	545
20.3	Suchen nach dem Hashverfahren	551
20.4	Suchen mit Hilfe von Backtracking	567
20.5	Übungsaufgaben	573

21	Präprozessor	577
21.1	Einfügen von Dateien	579
21.2	Symbolische Konstanten und Makros mit Parametern	581
21.3	Bedingte Compilierung	587
21.4	Informationen über den Übersetzungskontext	590
21.5	String-Umwandlungs- und Verknüpfungs-Operatoren	592
21.6	Präprozessor-Erweiterungen unter C99 und C11	594
22	Modulares Design in C	599
22.1	Konzept des Strukturierten Designs	599
22.2	Konzept des Modularen Designs	600
22.3	Umsetzung des Modularen Designs in C	606
22.4	Realisierung eines Stacks mit Modularem Design in C	611
22.5	Übungsaufgaben	619
23	Multithreading	621
23.1	Betriebssystemprozesse und Threads	621
23.2	Verwendung der Standard-Bibliothek <threads.h>	626
23.3	Synchronisation von Threads	633
23.4	Deadlocks, Livelocks und Starvation	651
23.5	Fortgeschrittene Programmierung mit Threads	653
A	Standardbibliotheksfunktionen	665
A.1	Klassifizierung und Konvertierung von Zeichen (ctype.h)	666
A.2	Mathematische Funktionen (math.h)	667
A.3	Ein- und Ausgabe (stdio.h)	668
A.4	Allgemeine Funktionen (stdlib.h)	671
A.5	String- und Speicherbearbeitung (string.h)	672
A.6	Datum und Uhrzeit (time.h)	673
B	Low-Level-Dateizugriffsfunktionen	675
B.1	Operationen auf dem Dateisystem	676
B.2	Ein- und Ausgabe, Positionieren innerhalb einer Datei	678
B.3	Beispiel zur Dateibearbeitung mit Low-Level-Funktionen	680
C	Wandlungen zwischen Zahlensystemen	683
C.1	Vorstellung der Zahlensysteme	683
C.2	Umwandlung von Dezimal in Binär/Hexadezimal	685
C.3	Umwandlung von Binär in Hexadezimal und umgekehrt	687
D	ASCII und Unicode	689
D.1	ASCII	690
D.2	Unicode	692

E	Arithmetische Typen, Wertebereiche und Codierungen	697
E.1	Modifikatoren short und long	697
E.2	Suffixe von arithmetischen Literalen	698
E.3	Wertebereiche von Integer-Typen und Gleitpunkt-Typen	699
E.4	Zweierkomplement-Codierung	701
E.5	Gleitpunkt-Codierung	702
Literatur		705
Index		707