

Inhalt

Vorwort	XXIII
Teil I: Grundkurs	1
1 Keine Angst vor C++!	3
1.1 Von C zu C++	4
1.1.1 Rückblick	4
1.1.2 Die strukturierte Programmierung	6
1.1.3 Chips sind billig, Programmierer teuer	8
1.1.4 Fassen wir zusammen	9
1.2 Von der Idee zum fertigen Programm	10
1.3 Näher hingeschaut: der C++-Compiler	12
1.3.1 Der Compiler ist ein strenger Lehrer	12
1.3.2 Definition und Deklaration	13
1.3.3 Das Konzept der Headerdateien	15
1.3.4 Namensräume	16
1.3.5 Der Compiler bei der Arbeit	18
1.3.6 ISO und die Compiler-Wahl	19
1.3.7 Der neue C++17-Standard	19
1.4 Übungen	20
2 Grundkurs: Das erste Programm	21
2.1 Hallo Welt! – das Programmgerüst	21
2.1.1 Typischer Programmaufbau	22
2.1.2 Die Eintrittsfunktion main()	23
2.1.3 Die Anweisungen	24
2.1.4 Headerdateien	26
2.1.5 Kommentare	27
2.2 Programmerstellung	28
2.2.1 Programmerstellung mit Visual Studio	28
2.2.2 Programmerstellung mit GNU-Compiler	35
2.2.3 Programmausführung	36

2.3	Stil	38
2.4	Übungen	39
3	Grundkurs: Daten und Variablen	41
3.1	Konstanten (Literele)	41
3.2	Variablen	44
3.2.1	Variablendefinition	44
3.2.2	Werte in Variablen speichern	47
3.2.3	Variablen bei der Definition initialisieren	48
3.2.4	Werte von Variablen abfragen	49
3.3	Konstante Variablen	50
3.4	Die Datentypen	51
3.4.1	Die Bedeutung des Datentyps	51
3.4.2	Die elementaren Datentypen	55
3.4.3	Weitere Datentypen	57
3.5	Typumwandlung	57
3.5.1	Typumwandlung bei der Ein- und Ausgabe	57
3.5.2	Automatische Typumwandlungen	60
3.5.3	Explizite Typumwandlungen	61
3.6	Übungen	62
4	Grundkurs: Operatoren und Ausdrücke	65
4.1	Rechenoperationen	65
4.1.1	Die arithmetischen Operatoren	65
4.1.2	Die mathematischen Funktionen	68
4.2	Ausdrücke	69
4.3	Die kombinierten Zuweisungen	71
4.4	Inkrement und Dekrement	71
4.5	Strings addieren	73
4.6	Weitere Operatoren	74
4.7	Übungen	74
5	Grundkurs: Kontrollstrukturen	75
5.1	Entscheidungen und Bedingungen	75
5.1.1	Bedingungen	76
5.1.2	Die Vergleichsoperatoren	77
5.1.3	Die logischen Operatoren	78
5.2	Verzweigungen	80
5.2.1	Die einfache if-Anweisung	80
5.2.2	Die if-else-Verzweigung	82
5.2.3	Die switch-Verzweigung	85

5.3	Schleifen	89
5.3.1	Die while-Schleife	89
5.3.2	Die do-while-Schleife	93
5.3.3	Die for-Schleife	95
5.3.4	Schleifen mit mehreren Schleifenvariablen	96
5.3.5	Performance-Tipps	97
5.4	Sprunganweisungen	97
5.4.1	Abbruchbefehle für Schleife	99
5.4.2	Abbruchbefehle für Funktionen	102
5.4.3	Sprünge mit goto	102
5.5	Fallstricke	102
5.5.1	Die leere Anweisung ;	102
5.5.2	Nebeneffekte in booleschen Ausdrücken	103
5.5.3	Dangling else-Problem	104
5.5.4	Endlosschleifen	105
5.6	Übungen	106
6	Grundkurs: Eigene Funktionen	109
6.1	Definition und Aufruf	110
6.1.1	Der Ort der Funktionsdefinition	111
6.1.2	Funktionsprototypen (Deklaration)	112
6.2	Rückgabewerte und Parameter	113
6.2.1	Rückgabewerte	115
6.2.2	Parameter	117
6.3	Lokale und globale Variablen	122
6.3.1	Lokale Variablen	122
6.3.2	Globale Variablen	123
6.3.3	Gültigkeitsbereiche und Verdeckung	124
6.4	Funktionen und der Stack	126
6.5	Überladung	128
6.6	Übungen	130
7	Grundkurs: Eigene Datentypen	131
7.1	Arrays	131
7.1.1	Definition	131
7.1.2	Auf Array-Elemente zugreifen	133
7.1.3	Initialisierung	133
7.1.4	Arrays in Schleifen durchlaufen	134
7.1.5	Arrays an Funktionen übergeben	137
7.1.6	Mehrdimensionale Arrays	137
7.1.7	Vor- und Nachteile der Programmierung mit Arrays	138

7.2	Aufzählungen	138
7.2.1	Definition	141
7.2.2	Variablen	141
7.2.3	Aufzählungstypen und switch-Verzweigungen	142
7.2.4	Die neuen enum class-Aufzählungen	142
7.3	Strukturen	143
7.3.1	Definition	144
7.3.2	Variablendefinition	145
7.3.3	Zugriff auf Elemente	146
7.3.4	Initialisierung	146
7.3.5	Arrays von Strukturen	146
7.4	Klassen	148
7.4.1	Definition	148
7.4.2	Variablen, Objekte und Konstruktoren	148
7.4.3	Zugriffsschutz	149
7.5	Übungen	152
8	Grundkurs: Zeiger und Referenzen	153
8.1	Zeiger	153
8.1.1	Definition	154
8.1.2	Initialisierung	154
8.1.3	Dereferenzierung	156
8.1.4	Zeigerarithmetik	158
8.2	Referenzen	159
8.3	Einsatzgebiete	159
8.3.1	call by reference	160
8.3.2	Dynamische Speicherreservierung	165
8.4	Übungen	171
9	Grundkurs: Noch ein paar Tipps	173
9.1	Wie gehe ich neue Programme an?	173
9.2	Wo finde ich Hilfe?	174
9.2.1	Hilfe zu Compiler-Meldungen	174
9.2.2	Hilfe bei der Lösung von Programmieraufgaben	175
9.2.3	Hilfe bei Programmen, die nicht richtig funktionieren	179
9.2.4	Debuggen	179
9.3	Programme optimieren	181

Teil II – Aufbaukurs: die Standardbibliothek	183
10 Aufbaukurs: Einführung	185
10.1 Bibliotheken verwenden	185
10.2 Hilfe zu den Bibliothekselementen	186
11 Aufbaukurs: Mathematische Funktionen	189
11.1 Die mathematischen Funktionen	189
11.1.1 Mathematische Konstanten	191
11.1.2 Verwendung der trigonometrischen Funktionen	192
11.1.3 Überläufe	192
11.2 Zufallszahlen	193
11.3 Komplexe Zahlen	195
11.4 Übungen	196
12 Aufbaukurs: Strings	197
12.1 String-Literale	197
12.1.1 Escape-Sequenzen	198
12.1.2 Zeilenumbrüche	200
12.2 Strings erzeugen	201
12.3 Strings aneinanderhängen	202
12.4 Strings vergleichen	202
12.5 Sonstige String-Manipulationen	205
12.6 C-Strings	206
12.7 Umwandlungen zwischen Strings und Zahlen	207
12.8 Übungen	208
13 Aufbaukurs: Ein- und Ausgabe	209
13.1 Daten auf die Konsole ausgeben	209
13.2 Formatierte Ausgabe	210
13.2.1 Ausgabebreite	210
13.2.2 Füllzeichen	211
13.2.3 Genauigkeit	211
13.2.4 Formatierte Ausgabe mit printf()	212
13.3 Deutsche Umlaute	213
13.4 Daten über die Konsole (Tastatur) einlesen	216
13.5 Fehlerbehandlung	217
13.6 Streams	219
13.7 Textdateien	220
13.7.1 In Textdateien schreiben	221
13.7.2 Aus Textdateien lesen	223

13.8	Binärdateien	226
13.9	Übungen	228
14	Aufbaukurs: Zeit und Datum	229
14.1	Zeit und Datum	229
14.2	Laufzeitmessungen	235
14.3	Übungen	237
15	Aufbaukurs: Container	239
15.1	Die STL	239
15.2	vector – ein intelligenter Daten-Container	242
15.2.1	Einsatz eines Containers	243
15.2.2	Größenmanagement von Containern	244
15.2.3	Typische Memberfunktionen	245
15.3	Der Gebrauch von Iteratoren	246
15.4	Die Algorithmen	249
15.4.1	generate()	252
15.4.2	stable_sort()	253
15.5	Schlüssel/Wert-Paare	254
15.6	Übungen	256
16	Aufbaukurs: Programme aus mehreren Quelltextdateien	257
16.1	Quelltext verteilen	257
16.1.1	Funktionen über Dateigrenzen hinweg verwenden	258
16.1.2	Klassen über Dateigrenzen hinweg verwenden	258
16.1.3	Variablen über Dateigrenzen hinweg verwenden	259
16.1.4	Typdefinitionen über Dateigrenzen hinweg verwenden	260
16.2	Mehrfacheinkopieren von Headerdateien verhindern	261
16.3	Übungen	263
Teil III – Objektorientierte Programmierung		265
17	OOP-Kurs: Klassen	267
17.1	Objektorientiert denken – objektorientiert programmieren	267
17.1.1	Objektorientiertes Programmieren	267
17.1.2	Wie sind Objekte beschaffen?	268
17.1.3	Wie findet man einen objektorientierten Lösungsansatz?	270
17.1.4	Objekte und Klassen	271
17.2	Klassendefinition	274
17.2.1	Zugriffsrechte	275

17.2.2	Quelltext- und Headerdatei	277
17.2.3	Klassen zu Visual-Studio-Projekten hinzufügen	280
17.3	Membervariablen	283
17.3.1	Anfangswerte	284
17.3.2	Private-Deklaration	288
17.3.3	Eingebettete Objekte	290
17.3.4	Konstante Membervariablen	292
17.3.5	Statische Membervariablen	293
17.4	Memberfunktionen	294
17.4.1	Definition innerhalb der Klassendefinition	294
17.4.2	Definition außerhalb der Klassendefinition	295
17.4.3	Der this-Zeiger	296
17.4.4	Statische Memberfunktionen	297
17.4.5	Konstante Memberfunktionen	298
17.4.6	Get/Set-Memberfunktionen	299
17.5	Die Konstruktoren	302
17.5.1	Definition und Aufruf	302
17.5.2	Ersatz- und Standardkonstruktoren	304
17.6	Der Destruktor	307
17.7	Übungen	308
18	OOP-Kurs: Vererbung	311
18.1	Das Prinzip der Vererbung	311
18.1.1	Der grundlegende Mechanismus	312
18.1.2	Die Syntax	313
18.1.3	Wann ist Vererbung gerechtfertigt?	314
18.1.4	Einige wichtige Fakten	315
18.2	Das Basisklassenunterobjekt	316
18.2.1	Zugriff	317
18.2.2	Instanzbildung	320
18.3	Die Zugriffsspezifizierer für die Vererbung	322
18.4	Verdecken, überschreiben und überladen	323
18.4.1	Verdeckung	324
18.4.2	Überladung	324
18.4.3	Überschreibung	325
18.5	Der Destruktor	325
18.6	Mehrfachvererbung	326
18.7	Übungen	326
19	OOP-Kurs: Polymorphie	329
19.1	Grundprinzip und Implementierung	330
19.2	Späte und frühe Bindung	333

19.2.1	Frühe Bindung	333
19.2.2	Späte Bindung	334
19.3	Generische Programmierung	335
19.3.1	Basisklassen-Arrays	336
19.3.2	Basisklassenparameter	338
19.4	Typidentifizierung zur Laufzeit (RTTI)	339
19.4.1	Umwandlung mit <code>dynamic_cast</code>	339
19.4.2	Der <code>typeid()</code> -Operator	341
19.5	Abstrakte Klassen	341
19.5.1	Rein virtuelle Funktionen	342
19.5.2	Abstrakte Klassen	342
19.6	Übungen	343
20	OOP-Kurs: Ausnahmebehandlung	345
20.1	Fehlerprüfung mit Ausnahmen	346
20.2	Ausnahmen abfangen	348
20.3	Ausnahmen auslösen	351
20.4	Programmfluss und Ausnahmebehandlung	353
20.4.1	Wo wird der Programmfluss nach einer Ausnahme fortgesetzt? ..	354
20.4.2	Die Problematik des gestörten Programmflusses	354
20.5	Übungen	356
Teil IV – Profikurs	357
21	Profikurs: Allgemeine Techniken	359
21.1	Vorzeichen und Überlauf	359
21.2	Arithmetische Konvertierungen	361
21.3	Lokale <code>static</code> -Variablen	361
21.4	Der <code>?:</code> -Operator	362
21.5	Bit-Operatoren	362
21.5.1	Multiplikation mit 2	363
21.5.2	Division durch 2	364
21.5.3	Klein- und Großschreibung	364
21.5.4	Flags umschalten	365
21.5.5	Gerade Zahlen erkennen	365
21.6	Zeiger auf Funktionen	367
21.7	Rekursion	369
21.8	<code>constexpr</code> -Funktionen	371
21.9	Variablendefinition in <code>if</code> und <code>switch</code>	372

22 Profikurs: Objektorientierte Techniken	375
22.1 Zeiger auf Memberfunktionen	375
22.2 Friends	377
22.3 Überladung von Operatoren	378
22.3.1 Syntax	378
22.3.2 Überladung des Inkrement-Operators ++	379
22.3.3 Überladung arithmetischer Operatoren +, +=	380
22.3.4 Überladung der Streamoperatoren <<>>	381
22.4 Objekte vergleichen	382
22.4.1 Gleichheit	382
22.4.2 Größenvergleiche	384
22.5 Objekte kopieren	386
23 Profikurs: Gültigkeitsbereiche und Lebensdauer	391
24 Profikurs: Templates	395
24.1 Funktionen-Templates	396
24.2 Klassen-Templates	397
25 Profikurs: Reguläre Ausdrücke	401
25.1 Syntax regulärer Ausdrücke	401
25.1.1 Zeichen und Zeichenklassen	402
25.1.2 Quantifizierer	403
25.1.3 Gruppierung	404
25.1.4 Assertionen (Anker)	405
25.2 Musterabgleich mit regulären Ausdrücken	405
25.3 Suchen mit regulären Ausdrücken	406
25.4 Ersetzen mit regulären Ausdrücken	407
26 Profikurs: Lambda-Ausdrücke	409
26.1 Syntax	409
26.2 Einsatz	411
A Anhang A: Lösungen	413
B Anhang B: Die Beispiele zum Buch	433
B.1 Installation der Visual Studio Community Edition	433
B.2 Ausführung der Beispielprogramme	436
B.2.1 Ausführung mit VS Community Edition 2017	437

B.2.2	Ausführung mit beliebigen integrierten Entwicklungsumgebungen	438
B.2.3	Ausführung mit GNU-Konsolen-Compiler	439
C	Anhang C: Zeichensätze	441
C.1	Der ASCII-Zeichensatz	441
C.2	Der ANSI-Zeichensatz	442
D	Anhang D: Syntaxreferenz	445
D.1	Schlüsselwörter	445
D.2	Elementare Typen	446
D.3	Strings	447
D.4	Operatoren	448
D.5	Ablaufsteuerung	449
D.6	Ausnahmebehandlung	451
D.7	Aufzählungen	451
D.7.1	enum	451
D.7.2	enum class (C++11)	452
D.8	Arrays	452
D.9	Zeiger	453
D.10	Strukturen	453
D.11	Klassen	454
D.12	Vererbung	457
D.13	Namensräume	457
E	Anhang E: Die Standardbibliothek	459
E.1	Die C-Standardbibliothek	459
E.2	Die C++-Standardbibliothek	460
Index	463