



# Inhalt

<b>Vorwort</b> .....	<b>XV</b>
<b>Teil I: Grundkurs</b> .....	<b>1</b>
<b>1 Keine Angst vor C++!</b> .....	<b>3</b>
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 .....	11
1.3.1 Der Compiler ist ein strenger Lehrer .....	12
1.3.2 Definition und Deklaration .....	13
1.3.3 Das Konzept der Headerdateien .....	14
1.3.4 Namensräume .....	15
1.3.5 Der Compiler bei der Arbeit .....	17
1.3.6 ISO und die Compiler-Wahl .....	18
1.3.7 Der neue C++11-Standard .....	19
1.4 Übungen .....	19
<b>2 Grundkurs: Das erste Programm</b> .....	<b>21</b>
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	34
2.2.3	Programmausführung	35
2.3	Stil	37
2.4	Übungen	38
<b>3</b>	<b>Grundkurs: Daten und Variablen</b>	<b>39</b>
3.1	Konstanten (Literele)	39
3.2	Variablen	42
3.2.1	Variablendefinition	42
3.2.2	Werte in Variablen speichern	45
3.2.3	Variablen bei der Definition initialisieren	46
3.2.4	Werte von Variablen abfragen	47
3.3	Konstante Variablen	48
3.4	Die Datentypen	49
3.4.1	Die Bedeutung des Datentyps	49
3.4.2	Die elementaren Datentypen	53
3.4.3	Weitere Datentypen	54
3.5	Typumwandlung	55
3.5.1	Typumwandlung bei der Ein- und Ausgabe	55
3.5.2	Automatische Typumwandlungen	57
3.5.3	Explizite Typumwandlungen	58
3.6	Übungen	60
<b>4</b>	<b>Grundkurs: Operatoren und Ausdrücke</b>	<b>61</b>
4.1	Rechenoperationen	61
4.1.1	Die arithmetischen Operatoren	61
4.1.2	Die mathematischen Funktionen	64
4.2	Ausdrücke	64
4.3	Die kombinierten Zuweisungen	66
4.4	Inkrement und Dekrement	67
4.5	Strings addieren	68
4.6	Weitere Operatoren	69
4.7	Übungen	70
<b>5</b>	<b>Grundkurs: Kontrollstrukturen</b>	<b>71</b>
5.1	Entscheidungen und Bedingungen	71
5.1.1	Bedingungen	72
5.1.2	Die Vergleichsoperatoren	73
5.1.3	Die logischen Operatoren	74
5.2	Verzweigungen	75
5.2.1	Die einfache if-Anweisung	76
5.2.2	Die if-else-Verzweigung	78
5.2.3	Die switch-Verzweigung	81

5.3	Schleifen	84
5.3.1	Die while-Schleife	84
5.3.2	Die do-while-Schleife	88
5.3.3	Die for-Schleife	89
5.3.4	Schleifen mit mehreren Schleifenvariablen	91
5.3.5	Performance-Tipps	91
5.4	Sprunganweisungen	92
5.4.1	Abbruchbefehle für Schleife	93
5.4.2	Abbruchbefehle für Funktionen	96
5.4.3	Sprünge mit goto	96
5.5	Fallstricke	97
5.5.1	Die leere Anweisung ;	97
5.5.2	Nebeneffekte in booleschen Ausdrücken	98
5.5.3	Dangling else-Problem	99
5.5.4	Endlosschleifen	100
5.6	Übungen	100
<b>6</b>	<b>Grundkurs: Eigene Funktionen</b>	<b>103</b>
6.1	Definition und Aufruf	104
6.1.1	Der Ort der Funktionsdefinition	105
6.1.2	Funktionsprototypen (Deklaration)	106
6.2	Rückgabewerte und Parameter	107
6.2.1	Rückgabewerte	109
6.2.2	Parameter	111
6.3	Lokale und globale Variablen	116
6.3.1	Lokale Variablen	116
6.3.2	Globale Variablen	116
6.3.3	Gültigkeitsbereiche und Verdeckung	118
6.4	Funktionen und der Stack	119
6.5	Überladung	121
6.6	Übungen	123
<b>7</b>	<b>Grundkurs: Eigene Datentypen</b>	<b>125</b>
7.1	Arrays	125
7.1.1	Definition	125
7.1.2	Auf Array-Elemente zugreifen	126
7.1.3	Initialisierung	127
7.1.4	Arrays in Schleifen durchlaufen	128
7.1.5	Arrays an Funktionen übergeben	131
7.1.6	Mehrdimensionale Arrays	131
7.1.7	Vor- und Nachteile der Programmierung mit Arrays	132
7.2	Aufzählungen	132
7.2.1	Definition	135

7.2.2	Variablen .....	135
7.2.3	Aufzählungstypen und switch-Verzweigungen .....	136
7.2.4	Die neuen enum class-Aufzählungen .....	136
7.3	Strukturen .....	137
7.3.1	Definition .....	138
7.3.2	Variablendefinition .....	139
7.3.3	Zugriff auf Elemente .....	139
7.3.4	Initialisierung .....	140
7.3.5	Arrays von Strukturen .....	140
7.4	Klassen .....	141
7.4.1	Definition .....	142
7.4.2	Variablen, Objekte und Konstruktoren .....	142
7.4.3	Zugriffsschutz .....	143
7.5	Übungen .....	145
<b>8</b>	<b>Grundkurs: Zeiger und Referenzen .....</b>	<b>147</b>
8.1	Zeiger .....	147
8.1.1	Definition .....	148
8.1.2	Initialisierung .....	148
8.1.3	Dereferenzierung .....	150
8.1.4	Zeigerarithmetik .....	152
8.2	Referenzen .....	152
8.3	Einsatzgebiete .....	153
8.3.1	call by reference .....	154
8.3.2	Dynamische Speicherreservierung .....	159
8.4	Übungen .....	165
<b>9</b>	<b>Grundkurs: Noch ein paar Tipps .....</b>	<b>167</b>
9.1	Wie gehe ich neue Programme an? .....	167
9.2	Wo finde ich Hilfe? .....	168
9.2.1	Hilfe zu Compiler-Meldungen .....	168
9.2.2	Hilfe bei der Lösung von Programmieraufgaben .....	170
9.2.3	Hilfe bei Programmen, die nicht richtig funktionieren .....	173
9.2.4	Debuggen .....	174
9.3	Programme optimieren .....	175
<b>Teil II: Aufbaukurs: die Standardbibliothek .....</b>		<b>177</b>
<b>10</b>	<b>Aufbaukurs: Einführung .....</b>	<b>179</b>
10.1	Bibliotheken verwenden .....	179
10.2	Hilfe zu den Bibliothekselementen .....	180

<b>11</b>	<b>Aufbaukurs: Mathematische Funktionen</b>	<b>183</b>
11.1	Die mathematischen Funktionen	183
11.1.1	Mathematische Konstanten	185
11.1.2	Verwendung der trigonometrischen Funktionen	186
11.1.3	Überläufe	186
11.2	Zufallszahlen	187
11.3	Komplexe Zahlen	189
11.4	Übungen	190
<b>12</b>	<b>Aufbaukurs: Strings</b>	<b>191</b>
12.1	String-Literale	191
12.1.1	Escape-Sequenzen	192
12.1.2	Zeilenumbrüche	194
12.2	Strings erzeugen	195
12.3	Strings aneinanderhängen	196
12.4	Strings vergleichen	196
12.5	Sonstige String-Manipulationen	199
12.6	C-Strings	200
12.7	Umwandlungen zwischen Strings und Zahlen	201
12.8	Übungen	202
<b>13</b>	<b>Aufbaukurs: Ein- und Ausgabe</b>	<b>203</b>
13.1	Daten auf die Konsole ausgeben	203
13.2	Formatierte Ausgabe	204
13.2.1	Ausgabebreite	204
13.2.2	Füllzeichen	205
13.2.3	Genauigkeit	205
13.2.4	Formatierte Ausgabe mit printf()	206
13.3	Deutsche Umlaute	207
13.4	Daten über die Konsole (Tastatur) einlesen	210
13.5	Fehlerbehandlung	211
13.6	Streams	213
13.7	Textdateien	214
13.7.1	In Textdateien schreiben	214
13.7.2	Aus Textdateien lesen	217
13.8	Binärdateien	220
13.9	Übungen	222
<b>14</b>	<b>Aufbaukurs: Zeit und Datum</b>	<b>223</b>
14.1	Zeit und Datum	223
14.2	Laufzeitmessungen	229
14.3	Übungen	231

<b>15</b>	<b>Aufbaukurs: Container</b>	<b>233</b>
15.1	Die STL	233
15.2	vector – ein intelligenter Daten-Container	236
15.2.1	Einsatz eines Containers	237
15.2.2	Größenmanagement von Containern	238
15.2.3	Typische Memberfunktionen	240
15.3	Der Gebrauch von Iteratoren	240
15.4	Die Algorithmen	243
15.4.1	generate()	246
15.4.2	stable_sort()	247
15.5	Schlüssel/Wert-Paare	248
15.6	Übungen	251
<b>16</b>	<b>Aufbaukurs: Programme aus mehreren Quelltextdateien</b>	<b>253</b>
16.1	Quelltext verteilen	253
16.1.1	Funktionen über Dateigrenzen hinweg verwenden	254
16.1.2	Klassen über Dateigrenzen hinweg verwenden	254
16.1.3	Variablen über Dateigrenzen hinweg verwenden	255
16.1.4	Typdefinitionen über Dateigrenzen hinweg verwenden	256
16.2	Mehrfacheinkopieren von Headerdateien verhindern	256
16.3	Übungen	258
<b>Teil III: Objektorientierte Programmierung</b>		<b>259</b>
<b>17</b>	<b>OOP-Kurs: Klassen</b>	<b>261</b>
17.1	Objektorientiert denken – objektorientiert programmieren	261
17.1.1	Objektorientiertes Programmieren	261
17.1.2	Wie sind Objekte beschaffen?	262
17.1.3	Wie findet man einen objektorientierten Lösungsansatz?	264
17.1.4	Objekte und Klassen	265
17.2	Klassendefinition	268
17.2.1	Zugriffsrechte	269
17.2.2	Quelltext- und Headerdatei	271
17.2.3	Klassen zu Visual-Studio-Projekten hinzufügen	274
17.3	Membervariablen	277
17.3.1	Anfangswerte	278
17.3.2	Private-Deklaration	282
17.3.3	Eingebettete Objekte	284
17.3.4	Konstante Membervariablen	286
17.3.5	Statische Membervariablen	287
17.4	Memberfunktionen	288
17.4.1	Definition innerhalb der Klassendefinition	288
17.4.2	Definition außerhalb der Klassendefinition	289

17.4.3	Der this-Zeiger .....	290
17.4.4	Statische Memberfunktionen .....	291
17.4.5	Konstante Memberfunktionen .....	292
17.4.6	Get-/Set-Memberfunktionen .....	293
17.5	Die Konstruktoren .....	296
17.5.1	Definition und Aufruf .....	296
17.5.2	Ersatz- und Standardkonstruktoren .....	298
17.6	Der Destruktor .....	301
17.7	Übungen .....	302
<b>18</b>	<b>OOP-Kurs: Vererbung .....</b>	<b>305</b>
18.1	Das Prinzip der Vererbung .....	305
18.1.1	Der grundlegende Mechanismus .....	306
18.1.2	Die Syntax .....	307
18.1.3	Wann ist Vererbung gerechtfertigt? .....	308
18.1.4	Einige wichtige Fakten .....	309
18.2	Das Basisklassenunterobjekt .....	310
18.2.1	Zugriff .....	311
18.2.2	Instanzbildung .....	314
18.3	Die Zugriffsspezifizierer für die Vererbung .....	316
18.4	Verdecken, überschreiben und überladen .....	317
18.4.1	Verdeckung .....	317
18.4.2	Überladung .....	318
18.4.3	Überschreibung .....	318
18.5	Der Destruktor .....	319
18.6	Mehrfachvererbung .....	319
18.7	Übungen .....	320
<b>19</b>	<b>OOP-Kurs: Polymorphie .....</b>	<b>323</b>
19.1	Grundprinzip und Implementierung .....	323
19.2	Späte und frühe Bindung .....	327
19.2.1	Frühe Bindung .....	327
19.2.2	Späte Bindung .....	328
19.3	Generische Programmierung .....	329
19.3.1	Basisklassen-Arrays .....	330
19.3.2	Basisklassenparameter .....	332
19.4	Typidentifizierung zur Laufzeit (RTTI) .....	333
19.4.1	Umwandlung mit dynamic_cast .....	333
19.4.2	Der typeid()-Operator .....	335
19.5	Abstrakte Klassen .....	335
19.5.1	Rein virtuelle Funktionen .....	336
19.5.2	Abstrakte Klassen .....	336
19.6	Übungen .....	337

<b>20</b>	<b>OOP-Kurs: Ausnahmebehandlung</b>	<b>339</b>
20.1	Fehlerprüfung mit Ausnahmen	340
20.2	Ausnahmen abfangen	342
20.3	Ausnahmen auslösen	345
20.4	Programmfluss und Ausnahmebehandlung	347
20.4.1	Wo wird der Programmfluss nach einer Ausnahme fortgesetzt?	347
20.4.2	Die Problematik des gestörten Programmflusses	348
20.5	Übungen	350
<b>Teil IV: Profikurs</b>		<b>351</b>
<b>21</b>	<b>Profikurs: Allgemeine Techniken</b>	<b>353</b>
21.1	Vorzeichen und Überlauf	353
21.2	Arithmetische Konvertierungen	355
21.3	Lokale static-Variablen	355
21.4	Der ?:-Operator	356
21.5	Bit-Operatoren	356
21.5.1	Multiplikation mit 2	358
21.5.2	Division durch 2	358
21.5.3	Klein- und Großschreibung	358
21.5.4	Flags umschalten	359
21.5.5	Gerade Zahlen erkennen	359
21.6	Zeiger auf Funktionen	361
21.7	Rekursion	363
<b>22</b>	<b>Profikurs: Objektorientierte Techniken</b>	<b>367</b>
22.1	Zeiger auf Memberfunktionen	367
22.2	Friends	369
22.3	Überladung von Operatoren	370
22.3.1	Syntax	370
22.3.2	Überladung des Inkrement-Operators ++	371
22.3.3	Überladung arithmetischer Operatoren +, +=	372
22.3.4	Überladung der Streamoperatoren <<>>	373
22.4	Objekte vergleichen	374
22.4.1	Gleichheit	374
22.4.2	Größenvergleiche	376
22.5	Objekte kopieren	378
<b>23</b>	<b>Profikurs: Gültigkeitsbereiche und Lebensdauer</b>	<b>383</b>



<b>24</b>	<b>Profikurs: Templates</b>	<b>387</b>
24.1	Funktionen-Templates	388
24.2	Klassen-Templates	389
<b>25</b>	<b>Profikurs: Reguläre Ausdrücke</b>	<b>393</b>
25.1	Syntax regulärer Ausdrücke	393
25.1.1	Zeichen und Zeichenklassen	394
25.1.2	Quantifizierer	395
25.1.3	Gruppierung	396
25.1.4	Assertionen (Anker)	397
25.2	Musterabgleich mit regulären Ausdrücken	397
25.3	Suchen mit regulären Ausdrücken	398
25.4	Ersetzen mit regulären Ausdrücken	399
<b>26</b>	<b>Profikurs: Lambda-Ausdrücke</b>	<b>401</b>
26.1	Syntax	401
26.2	Einsatz	403
<b>Anhang A: Lösungen</b>		<b>405</b>
<b>Anhang B: Die DVD zum Buch</b>		<b>425</b>
B.1	Installation von Visual Studio Express Edition	425
B.2	Ausführung der Beispielprogramme	427
B.2.1	Ausführung mit Visual Studio 2013 Express	427
B.2.2	Ausführung mit beliebigen integrierten Entwicklungsumgebungen	428
B.2.3	Ausführung mit GNU-Konsolen-Compiler	429
<b>Anhang C: Zeichensätze</b>		<b>431</b>
C.1	Der ASCII-Zeichensatz	431
C.2	Der ANSI-Zeichensatz	432
<b>Anhang D: Syntaxreferenz</b>		<b>435</b>
D.1	Schlüsselwörter	435
D.2	Elementare Typen	436
D.3	Strings	437
D.4	Operatoren	438
D.5	Ablaufsteuerung	440
D.6	Ausnahmebehandlung	441
D.7	Aufzählungen	442
D.7.1	enum	442
D.7.2	enum class (C++11)	442

D.8	Arrays .....	443
D.9	Zeiger .....	443
D.10	Strukturen .....	444
D.11	Klassen .....	444
D.12	Vererbung .....	447
<b>Anhang E: Die Standardbibliothek .....</b>		<b>449</b>
E.1	Die C-Standardbibliothek .....	449
E.2	Die C++-Standardbibliothek .....	450
<b>Index .....</b>		<b>453</b>