

## Auf einen Blick

1	Einstieg in die Welt von C++ .....	18
2	Erste Schritte in C++ .....	30
3	Die eingebauten C++-Basisdatentypen .....	44
4	Arbeiten mit den eingebauten Typen .....	68
5	Kontrollstrukturen .....	85
6	Arrays und Strings .....	120
7	Referenzen und Zeiger .....	138
8	Funktionen .....	150
9	Präprozessor-Direktiven und Modularisierung .....	181
10	Dynamische Speicherobjekte und Strukturen .....	212
11	Klassen .....	232
12	Objekte und Klassenelemente .....	275
13	Operatoren überladen .....	317
14	Vererbung (Abgeleitete Klassen) .....	345
15	Templates .....	366
16	Ausnahmebehandlung (Fehlerbehandlung) .....	390
17	Ein-/Ausgabestreams für Dateien .....	410
A	Weitere Features in C++11/C++14 .....	422
B	Lösungen der Übungsaufgaben .....	452

# Inhalt

<b>1</b>	<b>Einstieg in die Welt von C++</b>	<b>18</b>
<hr/>		
1.1	Der C++-Standard .....	18
1.2	Die nötigen Werkzeuge für C++ .....	19
1.3	Übersetzen mit einer Entwicklungsumgebung .....	22
1.4	Übersetzen mit g++ und clang++ .....	26
1.5	Listings zum Buch .....	28
1.6	Kontrollfragen und Aufgaben im Buch .....	28
1.7	Aufgabe .....	28
<b>2</b>	<b>Erste Schritte in C++</b>	<b>30</b>
<hr/>		
2.1	Das erste Programm in C++ .....	30
2.2	Anweisungen und Ausdrücke .....	32
2.2.1	Anweisungen .....	32
2.2.2	Ausdrücke .....	33
2.3	Die Standard-Eingabe- und -Ausgabestreams .....	34
2.3.1	Die Streams von C++ .....	34
2.3.2	Ausgabe mit »std::cout« .....	35
2.3.3	Ausgabe mit »std::cerr« oder »std::clog« .....	35
2.3.4	Eingabe mit »std::cin« .....	36
2.4	Einige Symbole von C++ .....	37
2.4.1	Bezeichner .....	38
2.4.2	Literale .....	38
2.4.3	Kommentare .....	42
2.5	Kontrollfragen und Aufgaben .....	43

<b>3</b>	<b>Die eingebauten C++-Basisdatentypen</b>	44
<b>3.1</b>	<b>Variablen</b>	44
<b>3.2</b>	<b>Deklaration und Definition von Variablen</b>	45
<b>3.3</b>	<b>Initialisierung und Zuweisung von Werten</b>	46
<b>3.4</b>	<b>Ganzzahltypen</b>	47
<b>3.5</b>	<b>Datentyp für Zeichen</b>	51
<b>3.6</b>	<b>Internationale Zeichen</b>	55
3.6.1	Unicode mit C++	55
3.6.2	Die Unicode-Typen »char16_t« und »char32_t«	56
3.6.3	Breite Zeichen – »wchar_t«	58
<b>3.7</b>	<b>Fließkommazahlentypen</b>	59
<b>3.8</b>	<b>Limits für die Basisdatentypen</b>	61
<b>3.9</b>	<b>Die Byte-Größe mit dem »sizeof«-Operator</b>	63
<b>3.10</b>	<b>Sicherheit beim Kompilieren mit »static_assert«</b>	63
<b>3.11</b>	<b>»auto«-Typ</b>	65
<b>3.12</b>	<b>Konstanten</b>	66
<b>3.13</b>	<b>Kontrollfragen und Aufgaben</b>	67
<b>4</b>	<b>Arbeiten mit den eingebauten Typen</b>	68
<b>4.1</b>	<b>Arithmetische Operatoren</b>	68
<b>4.2</b>	<b>Erweiterte Darstellung arithmetischer Operatoren</b>	72
<b>4.3</b>	<b>Inkrement- und Dekrementoperator</b>	72
<b>4.4</b>	<b>Ungenauere Fließkommazahlen</b>	75
<b>4.5</b>	<b>Typumwandlung</b>	77

4.5.1	Implizite Umwandlung durch den Compiler	77
4.5.2	Automatische Typumwandlung beschränken	80
4.5.3	Explizite Typumwandlung	81
<b>4.6</b>	<b>Kontrollfragen und Aufgaben</b>	84
<b>5</b>	<b>Kontrollstrukturen</b>	85
<b>5.1</b>	<b>Anweisungsblock für Kontrollstrukturen</b>	85
<b>5.2</b>	<b>Der eingebaute Datentyp »bool«</b>	87
<b>5.3</b>	<b>Bedingte Anweisung mit »if«</b>	88
<b>5.4</b>	<b>Vergleichsoperatoren</b>	92
<b>5.5</b>	<b>Alternative »else«-Verzweigung</b>	93
<b>5.6</b>	<b>Mehrfache Verzweigung</b>	95
<b>5.7</b>	<b>Der Bedingungsoperator »?:«</b>	98
<b>5.8</b>	<b>Logische Operatoren</b>	100
<b>5.9</b>	<b>Die Fallunterscheidung – »switch«</b>	103
<b>5.10</b>	<b>Die kopfgesteuerte »while«-Schleife</b>	106
<b>5.11</b>	<b>Die fußgesteuerte »do while«-Schleife</b>	109
<b>5.12</b>	<b>Die Zählschleife »for«</b>	112
<b>5.13</b>	<b>Kontrollierte Sprunganweisungen</b>	116
5.13.1	Die »break«-Anweisung	116
5.13.2	Die »continue«-Anweisung	117
<b>5.14</b>	<b>Kontrollfragen und Aufgaben</b>	119
<b>6</b>	<b>Arrays und Strings</b>	120
<b>6.1</b>	<b>Arrays</b>	120

6.1.1	Der C++-Container »std::vector« .....	121
6.1.2	C-Arrays .....	126
<b>6.2</b>	<b>Strings verwenden</b> .....	<b>131</b>
6.2.1	Der C++-Container »std::string« .....	131
6.2.2	Unterstützung von Unicode .....	133
6.2.3	C-Zeichenketten .....	134
<b>6.3</b>	<b>Kontrollfragen und Aufgaben</b> .....	<b>136</b>
<b>7</b>	<b>Referenzen und Zeiger</b> .....	<b>138</b>
<hr/>		
<b>7.1</b>	<b>Referenzen</b> .....	<b>138</b>
<b>7.2</b>	<b>Zeiger</b> .....	<b>141</b>
7.2.1	Syntax von Zeigern .....	142
7.2.2	Adresse im Zeiger speichern .....	143
7.2.3	Zeiger dereferenzieren .....	145
7.2.4	Zeiger prüfen .....	147
<b>7.3</b>	<b>Kontrollfragen und Aufgaben</b> .....	<b>148</b>
<b>8</b>	<b>Funktionen</b> .....	<b>150</b>
<hr/>		
<b>8.1</b>	<b>Grundlage zu den Funktionen</b> .....	<b>150</b>
8.1.1	Funktionen definieren .....	150
8.1.2	Funktionen aufrufen .....	151
8.1.3	Funktionen deklarieren .....	152
8.1.4	Funktionsparameter (Call-by-Value) .....	154
8.1.5	Konstante Funktionsparameter .....	156
8.1.6	Standardparameter .....	156
8.1.7	Rückgabewert von Funktionen .....	159
8.1.8	Funktionen überladen .....	161

8.1.9	Gültigkeitsbereich von lokalen und globalen Variablen .....	163
8.1.10	Die »main()«-Funktion .....	165
8.1.11	Programmende .....	167
<b>8.2</b>	<b>Referenzen als Parameter und Rückgabe</b> .....	<b>168</b>
8.2.1	Referenzen als Parameter .....	168
8.2.2	Referenzen als Rückgabe .....	172
<b>8.3</b>	<b>Zeiger als Parameter und Rückgabewert</b> .....	<b>173</b>
8.3.1	Referenzen vs. Zeiger als Parameter .....	174
<b>8.4</b>	<b>Größere Daten als Funktionsparameter</b> .....	<b>175</b>
<b>8.5</b>	<b>C-Arrays oder C-Strings als Funktionsparameter</b> .....	<b>177</b>
<b>8.6</b>	<b>Kontrollfragen und Aufgaben</b> .....	<b>178</b>
<b>9</b>	<b>Präprozessor-Direktiven und Modularisierung</b> .....	<b>181</b>
<hr/>		
<b>9.1</b>	<b>Präprozessor-Direktiven</b> .....	<b>181</b>
9.1.1	Die »#include«-Direktive .....	182
9.1.2	Die »#define«-Direktive .....	184
9.1.3	Bedingte Kompilierung .....	185
<b>9.2</b>	<b>Namensräume</b> .....	<b>187</b>
9.2.1	Neuen Namensbereich deklarieren .....	187
9.2.2	Namensbereich verschachteln .....	190
9.2.3	Namensbereich ist ein eigener Gültigkeitsbereich .....	190
9.2.4	Namensbereich mit »using« importieren .....	193
9.2.5	Einzelne Bezeichner mit »using« importieren .....	195
9.2.6	Aliasse für Namensbereiche .....	195
9.2.7	Anonymer Namensraum .....	196
9.2.8	Namensraum »std« .....	196
<b>9.3</b>	<b>Speicherklassenattribute</b> .....	<b>198</b>

9.3.1	Schlüsselwort »extern« .....	198
9.3.2	Schlüsselwort »static« .....	199
<b>9.4</b>	<b>Typqualifikatoren</b> .....	201
<b>9.5</b>	<b>Modularisierung</b> .....	202
9.5.1	Aufteilung .....	204
9.5.2	Die öffentliche Schnittstelle (Headerdatei) .....	205
9.5.3	Private Datei(en) .....	206
9.5.4	Die Client-Datei .....	207
9.5.5	Aufgabe .....	208
9.5.6	Nur Objektcode oder Bibliothek vorhanden .....	209
<b>9.6</b>	<b>Kontrollfragen und Aufgaben</b> .....	209
<b>10</b>	<b>Dynamische Speicherobjekte und Strukturen</b> .....	212
<b>10.1</b>	<b>Dynamische Speicherobjekte</b> .....	212
10.1.1	Dynamisch Objekte mit »new« anlegen .....	213
10.1.2	Fehler bei der Speichieranforderung abfangen .....	215
10.1.3	Speicher mit »delete« wieder freigeben .....	215
10.1.4	Der smarte »unique_ptr«-Pointer .....	217
<b>10.2</b>	<b>Erste eigene Datentypen mit Strukturen</b> .....	219
10.2.1	Strukturen deklarieren, erzeugen und initialisieren .....	221
10.2.2	Zugriff auf die Strukturelemente .....	222
10.2.3	Zugriff auf die Elemente in einer Funktion .....	223
10.2.4	Methoden statt Funktionen .....	225
10.2.5	Strukturen vergleichen .....	226
<b>10.3</b>	<b>Aufzählungstyp »enum«</b> .....	227
<b>10.4</b>	<b>Eigene Namen mit »using«</b> .....	230
<b>10.5</b>	<b>Kontrollfragen und Aufgaben</b> .....	231

<b>11</b>	<b>Klassen</b> .....	232
<b>11.1</b>	<b>Klassen</b> .....	232
11.1.1	Klassendefinition .....	233
11.1.2	Methoden definieren .....	235
11.1.3	Zugriffskontrolle mit »public« und »private« .....	237
11.1.4	Zugriff auf die Daten innerhalb einer Klasse .....	240
11.1.5	Objekte erzeugen und benutzen .....	241
<b>11.2</b>	<b>Konstruktoren</b> .....	246
11.2.1	Konstruktoren deklarieren .....	247
11.2.2	Konstruktoren definieren .....	248
11.2.3	Implizite Konvertierungen verhindern – »explicit« .....	251
11.2.4	Optimierung 1: Klasselemente gleich direkt initialisieren .....	252
11.2.5	Optimierung 2: Konstruktoren delegieren .....	253
11.2.6	Standardkonstruktor (Default-Konstruktor) .....	255
11.2.7	Kopierkonstruktor .....	256
11.2.8	Verschiebekonstruktor (Move-Konstruktor) .....	258
<b>11.3</b>	<b>Destruktoren</b> .....	260
11.3.1	Lebensdauer eines Objekts .....	261
11.3.2	Wann wird ein Destruktor erforderlich? .....	261
11.3.3	Destruktor deklarieren .....	262
11.3.4	Destruktor definieren .....	263
<b>11.4</b>	<b>Methoden</b> .....	266
11.4.1	»inline«-Methoden .....	266
11.4.2	Konstante Methoden (»nur-lesen-erlaubt«) .....	269
11.4.3	»this«-Zeiger .....	271
<b>11.5</b>	<b>Kontrollfragen und Aufgaben</b> .....	273

<b>12 Objekte und Klasselemente</b>	275
<b>12.1 Konstante Objekte</b>	275
<b>12.2 Objekt einer Klasse als Parameter</b>	276
12.2.1 Objekte einer Klasseninstanz an eine Funktion übergeben	276
12.2.2 Objekte von Klasseninstanz an eine Methode übergeben	280
12.2.3 Das Zielobjekt mit dem »this«-Zeiger	282
<b>12.3 Objekte einer Klasse als Rückgabewerte</b>	284
<b>12.4 Arrays von Objekten</b>	287
<b>12.5 Dynamische Objekte</b>	288
<b>12.6 Mehr zu den Klasselementen</b>	290
12.6.1 Rohe Zeiger als Klasselemente oder direkt die Nullregel	290
12.6.2 Die großen Fünf	299
12.6.3 Statische Klasselemente	302
12.6.4 Konstante Klasselemente	306
<b>12.7 Andere Klassenobjekte als Datenelement einer Klasse</b>	308
<b>12.8 Freundfunktionen (»friend«)</b>	313
<b>12.9 Kontrollfragen und Aufgaben</b>	315
<b>13 Operatoren überladen</b>	317
<b>13.1 Schlüsselwort »operator«</b>	318
<b>13.2 Zweistellige (arithmetische) Operatoren überladen</b>	320
13.2.1 Operatorüberladung als Methode einer Klasse	321
13.2.2 Operatorüberladung als globale Hilfsfunktion	325

<b>13.3 Einstellige Operatoren überladen</b>	327
<b>13.4 Zuweisungsoperator überladen</b>	331
<b>13.5 Ein-/Ausgabeoperator überladen</b>	336
13.5.1 Eingabeoperator »>>« überladen	336
13.5.2 Ausgabeoperator »<<« überladen	337
<b>13.6 Weitere Operatorüberladungen</b>	339
<b>13.7 Konvertierungsoperatoren</b>	340
13.7.1 Konvertierungskonstruktor	340
13.7.2 Konvertierungsfunktion	342
<b>13.8 Kontrollfragen und Aufgaben</b>	344
<b>14 Vererbung (Abgeleitete Klassen)</b>	345
<b>14.1 Die Vorbereitung</b>	346
<b>14.2 Die Ableitung einer Klasse</b>	347
14.2.1 »public«-Zugriffsrechte einer abgeleiteten Klasse	350
14.2.2 Erben und erweitern	350
14.2.3 Zugriff auf die Daten	351
14.2.4 Überschreiben von Klasselementen	353
14.2.5 Konstruktoren	354
14.2.6 Destruktor	356
14.2.7 Programmbeispiel	356
14.2.8 Zugriffsrecht »protected«	357
14.2.9 Implizite Typumwandlung abgeleiteter Klassen	359
14.2.10 Konstruktoren vererben	360
14.2.11 Virtuelle Methoden	361
14.2.12 Abstrakte Klassen und Methoden	364
<b>14.3 Kontrollfragen und Aufgaben</b>	364

<b>15 Templates</b>	366
<b>15.1 Funktions-Templates</b>	366
15.1.1 Funktions-Template definieren	367
15.1.2 Typübereinstimmung	370
15.1.3 Funktions-Templates über mehrere Module	370
15.1.4 Funktions-Template spezialisieren	370
15.1.5 Templates mit verschiedenen Parametern	372
15.1.6 Explizite Template-Argumente	374
<b>15.2 Klassen-Templates</b>	375
15.2.1 Klassen-Template definieren	375
15.2.2 Methoden von Klassen-Templates definieren	376
15.2.3 Methoden von Klassen-Template spezialisieren	377
15.2.4 Klassen-Template instanziiieren	378
15.2.5 Klassen-Template mit mehreren formalen Parametern	380
<b>15.3 Templates der Standardbibliothek</b>	380
15.3.1 Container der Standardbibliothek	381
15.3.2 Iteratoren	386
15.3.3 Algorithmen	386
<b>15.4 Kontrollfragen und Aufgaben</b>	388
<b>16 Ausnahmebehandlung (Fehlerbehandlung)</b>	390
<b>16.1 Ausnahme auslösen</b>	391
<b>16.2 Ausnahme auffangen und behandeln</b>	392
16.2.1 Alternatives »catch (...)«	396
16.2.2 Stack-Abwicklung	397
16.2.3 Ausnahme mit »throw« weiterwerfen	397
<b>16.3 Ausnahmeklassen (Fehlerklassen)</b>	400

<b>16.4 Standardausnahmen</b>	402
16.4.1 Virtuelle Methode »what()«	403
16.4.2 Anwenden der Standardausnahmen	403
<b>16.5 Systemausnahmen</b>	407
16.5.1 »bad_alloc«	407
16.5.2 »bad_cast«	408
16.5.3 »bad_typeid«	408
16.5.4 »bad_exception«	408
<b>16.6 Kontrollfragen</b>	409
<b>17 Ein-/Ausgabestreams für Dateien</b>	410
<b>17.1 Der Umgang mit Dateien in C++</b>	410
<b>17.2 Verschiedene Streams für Dateien</b>	411
<b>17.3 Datei öffnen und schließen</b>	411
17.3.1 Verschiedene Modi zum Öffnen von Dateien	412
17.3.2 Datei schließen	413
17.3.3 Formatiertes Lesen und Schreiben	414
17.3.4 Byteweise lesen und schreiben	415
17.3.5 Zeilenweise lesen und schreiben	417
17.3.6 Blockweise lesen und schreiben	418
17.3.7 Lese- oder Schreibeposition ändern	419
<b>17.4 Kontrollfragen und Aufgaben</b>	421
<b>Anhang</b>	422
<b>A Weitere Features in C++11/C++14</b>	422
A.1 Grundsätzliche Neuerungen in der Kernsprache	422
A.1.1 Rangebasierte »for«-Schleife	422

A.1.2	Lambda-Funktionen .....	423
A.1.3	RValue (neue Move-Semantik) .....	425
A.1.4	Generische Programmierung – Variadic Templates .....	426
A.1.5	»decltype« und die neue Rückgabesyntax .....	428
A.1.6	»constexpr« .....	431
A.1.7	Code mit »[[deprecated]]« als veraltet kennzeichnen (C++14) .....	432
A.1.8	Binäres Literal (C++14) .....	434
A.1.9	Digitales Trennzeichen für große Zahlen (C++14) .....	434
A.1.10	Automatische Rückgabetypermittlung (C++14) .....	435
A.2	Standardbibliothek – neue Bibliotheken .....	435
A.2.1	Reguläre Ausdrücke .....	436
A.2.2	Zeitbibliothek .....	438
A.2.3	Smart Pointer .....	441
A.3	Multithreading .....	445
A.3.1	Einfache Threads erzeugen .....	446
A.3.2	Chaos vermeiden .....	448
<b>B</b>	<b>Lösungen der Übungsaufgaben</b> .....	<b>452</b>
B.1	Lösungen zu Kapitel 2 .....	452
B.2	Lösungen zu Kapitel 3 .....	452
B.3	Lösungen zu Kapitel 4 .....	453
B.4	Lösungen zu Kapitel 5 .....	455
B.5	Lösungen zu Kapitel 6 .....	456
B.6	Lösungen zu Kapitel 7 .....	457
B.7	Lösungen zu Kapitel 8 .....	457
B.8	Lösungen zu Kapitel 9 .....	460
B.9	Lösungen zu Kapitel 10 .....	462
B.10	Lösungen zu Kapitel 11 .....	464
B.11	Lösungen zu Kapitel 12 .....	465

B.12	Lösungen zu Kapitel 13 .....	467
B.13	Lösungen zu Kapitel 14 .....	468
B.14	Lösungen zu Kapitel 15 .....	470
B.15	Lösungen zu Kapitel 16 .....	471
B.16	Lösungen zu Kapitel 17 .....	471
Index	.....	473