

# ***Inhaltsverzeichnis***

<b><i>Einführung</i></b>	<b><i>19</i></b>
Zu diesem Buch	19
Über die CD	19
Was ist C++?	20
Konventionen in diesem Buch	20
Wie dieses Buch aufgebaut ist	21
Das ist noch nicht alles	21
Teil I: Einführung in die C++-Programmierung	22
Teil II: Ein brauchbarer C++-Programmierer werden	22
Teil III: Eine Einführung in Klassen	22
Teil IV: Vererbung	22
Teil V: Optionale Funktionen	22
Teil VI: Der Top-Ten-Teil	23
Icons in diesem Buch	23
Wie geht's weiter?	23
<b><i>Teil I</i></b>	
<b><i>Einführung in die C++-Programmierung</i></b>	<b><i>25</i></b>
<b><i>Kapitel 1</i></b>	
<b><i>Ihr erstes C++-Programm</i></b>	<b><i>27</i></b>
Was ist ein Programm?	27
Code::Blocks unter Windows installieren	28
Ihr erstes C++-Programm erstellen	31
Ein Projekt anlegen	32
Den C++-Code eingeben	33
Ihr Programm erstellen	35
Ihr Programm ausführen	36
Kommentierte Programme lesen	37
Das gemeinsame Gerüst aller C++-Programme	38
Sourcecode durch Kommentare erläutern	38
Programme bestehen aus C++-Anweisungen	39
Deklarationen schreiben	40
Ausgaben erzeugen	41
Ausdrücke berechnen	41
Das Ergebnis eines Ausdrucks speichern	41
Den Rest des Programms untersuchen	42

## ***C++ für Dummies***

### ***Kapitel 2***

#### ***Variablen verwenden***

**43**

Variablen deklarieren	43
Verschiedene Variablentypen für alle Fälle	44
Überblick über die Einschränkungen der Ganzzahlen in C++	45
Das Rundungsproblem lösen	46
Die Einschränkungen von Fließkommazahlen	47
Variablentypen deklarieren	48
Konstantentypen	50
Wertebereiche von numerischen Typen	51
Sonderzeichen	52
Zeichen mit doppelter Spurbreite	53
Sind diese Berechnungen wirklich logisch?	54
Mixed-Mode-Ausdrücke	54
Automatische Deklarationen	55

### ***Kapitel 3***

#### ***Mathematische Operationen ausführen***

**57**

Einfache binäre Operationen ausführen	57
Ausdrücke zerlegen	59
Die Reihenfolge von Operationen bestimmen	59
Unäre Operationen ausführen	60
Mit Zuweisungsoperatoren arbeiten	61

### ***Kapitel 4***

#### ***Logische Operationen ausführen***

**63**

Warum brauchen wir überhaupt logische Operationen?	63
Mit den einfachen logischen Operatoren arbeiten	64
Logische Werte speichern	65
Mit logischen int-Variablen arbeiten	66
Vorsicht bei logischen Operationen mit Fließkommavariablen	67
Binärzahlen repräsentieren	68
Das dezimale Zahlensystem	69
Andere Zahlensysteme	69
Das binäre Zahlensystem	69
Bitweise logische Operationen ausführen	70
Die bitweisen Operatoren	71
Die bitweisen Operatoren verwenden	72
Ein einfacher Test	73
Etwas Logisches mit logischen Berechnungen tun	75

<b>Kapitel 5</b>	
<b>Den Programmablauf steuern</b>	<b>77</b>
Den Programmablauf mit Verzweigungsbefehlen steuern	77
Schleifen in einem Programm ausführen	79
Eine Schleife ausführen, während eine Bedingung true ist	80
Die Autoinkrement- und Autodekrement-Funktionen verwenden	81
Mit der for-Schleife arbeiten	83
Die gefürchtete Endlosschleife vermeiden	85
Spezielle Schleifenkontrollen anwenden	86
Steuerbefehle verschachteln	89
Die Mehrwege-Verzweigung: switch	91
<b>Teil II</b>	
<b>Ein brauchbarer C++-Programmierer werden</b>	<b>93</b>
<b>Kapitel 6</b>	
<b>Funktionen erstellen</b>	<b>95</b>
Eine Funktion schreiben und verwenden	95
Unsere erste Funktion definieren	98
Die Funktion sumSequence() definieren	98
Die Funktion sumSequence() aufrufen	98
Teile und herrsche	99
Die Details von Funktionen verstehen	99
Einfache Funktionen verstehen	100
Funktionen mit Argumenten verstehen	100
Funktionsnamen überladen	104
Funktionsprototypen definieren	105
Variable Speichertypen	107
<b>Kapitel 7</b>	
<b>Sequenzen in Arrays speichern</b>	<b>109</b>
Wozu werden Arrays benötigt?	109
Ein Array verwenden	110
Ein Array initialisieren	114
Mit einem zu großen Indexwert auf ein Array zugreifen	115
Mit Arrays arbeiten	115
Arrays von Arrays definieren und verwenden	115
Arrays von Zeichen verwenden	116
Ein Array von Zeichen erstellen	116
Einen String von Zeichen erstellen	117
Zeichenstrings manipulieren	119
Ein paar Bibliotheksfunktionen	121
Mehr Platz für Wide-Strings	122

## **C++ für Dummies**

### **Kapitel 8**

#### **Ein erster Blick auf C++-Zeiger**

**125**

Variablengröße	125
Was ist in einer Adresse enthalten?	127
Adressoperatoren	127
Zeigervariablen verwenden	129
Verschiedene Typen von Zeigern verwenden	130
Zeiger an Funktionen übergeben	131
Übergabe als Wert	131
Zeigerwert übergeben	132
Übergabe per Referenz	133
Konstante Irritationen	134
Einen Speicherblock namens Heap nutzen	135
Begrenzter Gültigkeitsbereich	135
Das Problem des Gültigkeitsbereichs untersuchen	137
Mit Hilfe des Heaps eine Lösung finden	137

### **Kapitel 9**

#### **Ein zweiter Blick auf C++-Zeiger**

**139**

Operationen für Zeigervariablen definieren	139
Arrays im Licht von Zeigervariablen noch einmal betrachten	140
Operatoren auf die Adresse eines Arrays anwenden	141
Zeigeroperationen auf einen String ausdehnen	143
Gründe für die zeigerbasierte String-Manipulation	144
Operatoren auf andere Zeigertypen als char anwenden	145
Einen Zeiger mit einem Array vergleichen	145
Wann gibt's einen Zeiger nicht?	148
Arrays von Zeigern deklarieren und verwenden	149
Arrays von Zeichenstrings nutzen	150
Auf die Argumente von main() zugreifen	151

### **Kapitel 10**

#### **Der C++-Präprozessor**

**155**

Was ist ein Präprozessor?	155
Einbinden von Dateien	156
Die #Definition von Dingen	159
Und wenn ich nichts #definieren will?	161
Ein paar Alternativen aufzählen	162
Dinge mit einem »ja aber« einbinden	163
Vordefinierte Objekte	164
Typendefinitionen mit typedef	167

<b>Teil III</b>	
<b><i>Eine Einführung in Klassen</i></b>	<b>169</b>
<b><i>Kapitel 11</i></b>	
<b><i>Ein Blick auf die objektorientierte Programmierung</i></b>	<b>171</b>
Die Abstraktion und der Mikrowellenherd	171
Nachos prozedural zubereiten	172
Nachos objektorientiert zubereiten	172
Klassenbildung und Mikrowellen	173
Wozu Klassen bilden?	174
<b><i>Kapitel 12</i></b>	
<b><i>Klassen in C++</i></b>	<b>175</b>
Einführung der Klasse	175
Das Format einer Klasse	175
Auf die Elemente einer Klasse zugreifen	176
Objekte aktivieren	177
Reale Objekte simulieren	177
Wozu dienen Elementfunktionen?	178
Eine Elementfunktion hinzufügen	179
Eine Elementfunktion aufrufen	180
Von einer Elementfunktion auf andere Elemente zugreifen	182
Auflösung des Gültigkeitsbereichs	183
Eine Elementfunktion in der Klasse definieren	184
Elementfunktionen separat definieren	186
Elementfunktionen überladen	188
<b><i>Kapitel 13</i></b>	
<b><i>Objekte mit Zeigern manipulieren</i></b>	<b>191</b>
Arrays aus Objekten deklarieren	191
Zeiger auf Objekte deklarieren	192
Einen Objektzeiger dereferenzieren	193
Der Pfeiloperator	194
Objekte an Funktionen übergeben	194
Eine Funktion mit einem Objektwert aufrufen	194
Eine Funktion mit einem Objektzeiger aufrufen	196
Eine Funktion mit dem Referenzoperator aufrufen	197
Wozu Zeiger oder Referenzen?	199
Zurück zum Heap	199
Viele Objekte auf dem Heap anlegen	200
Zeiger und Referenzen vergleichen	201
Verknüpfungen mit verketteten Listen erstellen	201
Andere Operationen auf eine verkettete Liste anwenden	202
Das LinkedListData-Beispielprogramm	203

## ***C++ für Dummies***

Ein Hoffungsstrahl: Eine Liste von Containern, die mit der C++-Library verknüpft sind	207
---	-----

### ***Kapitel 14***

#### ***Nicht stören: Elemente schützen*** **209**

Elemente schützen	209
Wozu benötigen Sie geschützte Elemente?	209
Wie geschützte Elemente funktionieren	210
Argumente für den Einsatz geschützter Elemente	211
Den internen Status einer Klasse schützen	211
Eine Klasse mit eingeschränkter Schnittstelle verwenden	212
Nicht-Elementfunktionen Zugriff auf geschützte Elemente geben	213

### ***Kapitel 15***

#### ***Objekte konstruieren und zerstören*** **217**

Objekte erstellen	217
Konstruktoren verwenden	218
Ein einzelnes Objekt konstruieren	218
Mehrere Objekte konstruieren	220
Ein Duplex konstruieren	221
Einen Destruktor analysieren	223
Warum Sie den Destruktor benötigen	223
Mit Destrukturen arbeiten	223

### ***Kapitel 16***

#### ***Konstruktoren mit Argumenten*** **229**

Konstruktoren mit Argumenten ausstatten	229
Einen Konstruktor verwenden	230
Konstruktoren überladen	232
Standardmäßige Standardkonstruktoren	234
Klassenelemente konstruieren	236
Ein komplexes Datenelement konstruieren	236
Ein konstantes Datenelement konstruieren	241
Die Reihenfolge der Konstruktion	241
Lokale Objekte werden der Reihenfolge nach konstruiert	242
Statische Objekte werden nur einmal konstruiert	242
Alle globalen Objekte werden vor main() konstruiert	243
Globale Objekte werden in keiner bestimmten Reihenfolge konstruiert	243
Elemente werden in der Reihenfolge konstruiert, in der sie deklariert werden	245
Destrukturen werden in der umgekehrten Reihenfolge der	
Konstruktoren aufgerufen	245
Konstruktoren als Form der Konvertierung	245

<b>Kapitel 17</b>	
<b>Der Copy-Konstruktor</b>	<b>247</b>
Ein Objekt kopieren	247
Warum Sie Copy-Konstruktoren brauchen	247
Die Copy-Konstruktoren verwenden	248
Der automatische Copy-Konstruktor	250
Flache oder tiefe Kopien erstellen	252
Temporäre Objekte	256
Temporäre Objekte dauerhaft vermeiden	258
<b>Kapitel 18</b>	
<b>Statische Elemente</b>	<b>259</b>
Ein statisches Element definieren	259
Warum Sie statische Elemente benötigen	259
Statische Elemente verwenden	260
Statische Datenelemente referenzieren	261
Verwendungsmöglichkeiten statischer Datenelemente	262
Statische Elementfunktionen deklarieren	263
Was ist überhaupt this?	265
<b>Teil IV</b>	
<b>Vererbung</b>	<b>267</b>
<b>Kapitel 19</b>	
<b>Eine Klasse erben</b>	<b>269</b>
Brauche ich Vererbung?	270
Wie funktioniert die Vererbung einer Klasse?	271
Eine Unterklasse verwenden	273
Eine Unterklasse konstruieren	273
Eine Unterklasse zerstören	275
Die HAS_A-Beziehung	275
<b>Kapitel 20</b>	
<b>Virtuelle Elementfunktionen</b>	<b>277</b>
Warum Sie Polymorphismus brauchen	280
Wie funktioniert Polymorphismus?	280
Wann ist eine Funktion virtuell und wann nicht?	282
Virtuelle Betrachtungen	283

**Kapitel 21**

**Klassen faktorisieren**

**285**

Das Faktorisieren

285

Abstrakte Klassen implementieren

289

Das Konzept der abstrakten Klasse

290

Wie man aus einer abstrakten Klasse eine wirkliche Klasse macht

292

Abstrakte Klassen übergeben

292

**Teil V**

**Optionale Funktionen**

**293**

**Kapitel 22**

**Ein neuer Zuweisungsoperator**

**295**

Operatoren mit Funktionen vergleichen

295

Einen neuen Operator einfügen

296

Flache Kopien zu erstellen ist ein tiefes Problem

296

Den Zuweisungsoperator überladen

298

Den Indexoperator überladen

303

**Kapitel 23**

**Mit Stream-I/O arbeiten**

**305**

Wie funktioniert Stream-I/O?

305

Default Stream-Objekte

306

Die fstream-Unterklassen

307

Vielfältiges Öffnen

308

Datei, was machst Du gerade?

309

Kann ich ein Beispiel sehen?

309

Andere Methoden der Stream-Klassen

313

Streams direkt lesen und schreiben

314

Das Format kontrollieren

316

Was ist mit endl?

318

Positionierung des Zeigers innerhalb einer Datei

318

Die stringstream-Unterklassen verwenden

319

Manipulatoren manipulieren

322

**Kapitel 24**

**Fehler verarbeiten – Ausnahmen**

**325**

Warum brauche ich einen neuen Fehlermechanismus?

327

Den Ausnahmemechanismus untersuchen

328

Welche Dinge sind »throw«-fähig?

331

Einfach mal durchreichen

334



<b>Kapitel 25</b>	
<b><i>Mehrfachvererbung</i></b>	<b>337</b>
Wie funktioniert die Mehrfachvererbung?	337
Vererbungsmehrdeutigkeiten beseitigen	339
Virtuelle Vererbung hinzufügen	340
Objekte der Mehrfachvererbung erzeugen	346
Eine gegenteilige Meinung	347
 <b>Kapitel 26</b>	
<b><i>C++-Templates durchleuchten</i></b>	<b>349</b>
Eine Funktion in einem Template verallgemeinern	350
Template-Klassen	352
Tipps für die Verwendung von Templates	356
 <b>Kapitel 27</b>	
<b><i>Die Standard Template Library als Standard verwenden</i></b>	<b>357</b>
Der string-Container	358
Die list-Container	362
Listen durchwandern	363
Operationen auf ganzen Listen	364
Beispiel, bitte!	365
 <b>Teil VI</b>	
<b><i>Der Top-Ten-Teil</i></b>	<b>369</b>
 <b>Kapitel 28</b>	
<b><i>Zehn Methoden, um Fehler in einem Programm zu vermeiden</i></b>	<b>371</b>
Aktivieren Sie alle Warnungen und Fehlermeldungen	371
Pflegen Sie einen sauberen und konsistenten Programmierstil	372
Schränken Sie die Sichtbarkeitsbereiche ein	372
Kommentieren Sie Ihren Code	374
Durchlaufen Sie jeden Pfad mindestens einmal im Einzelschrittmodus	374
Vermeiden Sie es, Operatoren zu überladen	374
Mit dem Heap arbeiten	375
Fehler mit Ausnahmen behandeln	375
Deklarieren Sie Destruktoren virtuell	375
Vermeiden Sie die Mehrfachvererbung	377

## ***C++ für Dummies***

### ***Kapitel 29***

#### ***(Fast) Zehn Websites zu allen Fragen rund um C++***

**379**

C-plusplus.de (deutsch)	379
Cplusplus.com (englisch)	379
Code::Blocks	379
Visual C++ Express Edition	380
Forum zur Visual C++ Express Edition	380
Referenz zur Windows-Programmierung mit C++	380
Referenz zu C (englisch)	380
Referenz zu C (deutsch)	380
Informationen zu C++11	380

### ***Anhang A***

#### ***Über die CD***

**381**

Systemanforderungen	381
Die CD benutzen	381
Der Inhalt der CD	382
Entwicklungswerkzeug	382
Programm-Quellcode	382
Falls Sie Probleme mit der CD haben	382

### ***Stichwortverzeichnis***

**385**