

Inhaltsverzeichnis

1	Willkommen in „Down Under“	17
1.1	Warum dieses Buch?	18
1.2	Für wen?	18
1.3	Was erwartet Sie?	18
1.4	Was erwartet Sie nicht?	19
1.5	Aufbau des Buches	19
	Aufbau der Kapitel.....	20
	Verwendete Notation	21
	Hervorhebungen.....	21
1.6	Was ist anders in der 3. Auflage?	22
1.7	Wo endet das Buch?	22
2	Einführung in die Objektorientierung	23
2.1	Motivation	24
2.2	Wie alles begann	24
2.3	Was ist OO?	25
2.4	Vererbungslehre	25
2.5	Mehrfach erben	26
2.6	Was man sonst noch über OO wissen sollte.....	27
2.7	Und wie sieht die konventionelle Programmierung aus?	28

2.8	Aufgaben.....	30
2.9	Zusammenfassung	30
3	Die ersten Schritte.....	31
3.1	Motivation	32
3.2	Ein C++-Programm entsteht	32
	Die Entwurfsphase.....	32
	Der Editor	33
	Das Übersetzen.....	33
3.3	Hello World.....	35
3.4	Die Eclipse-Plattform	36
	Aufruf des Compilers.....	40
	Syntax-Error	42
	Der Programmstart.....	43
	Arbeit sichern	45
3.5	Die Fehlersuche.....	45
3.6	Aufgaben.....	45
3.7	Zusammenfassung	46
4	Der Präprozessor	47
4.1	Motivation	48
4.2	Kommentare.....	48
4.3	Textersetzung und Makrodefinition	49
	Textersetzung	49
	Makrodefinition.....	50
	Der Operator „#“	52
	Der Operator „##“	52
	Makros aufheben	52
4.4	Header-Dateien.....	53

4.5	Bedingte Übersetzung	54
	#ifdef	54
	#ifndef	54
	#if	55
	#else, #elif	56
	#error	56
4.6	Sonstige Aufgaben	57
	Ersatzdarstellung auflösen.....	57
	Verknüpfen von Zeilen.....	58
	Benachbarten Text zusammenfassen.....	59
	#pragma	59
4.7	Aufruf des Präprozessors	59
4.8	Vordefinierte Namen	60
4.9	Aufgaben.....	61
4.10	Zusammenfassung.....	62
5	Einfache Anweisungen.....	63
5.1	Motivation	64
5.2	Was sind Anweisungen?.....	64
5.3	Ein-/Ausgabe.....	66
	Ausgabe.....	66
	Ausgabe von Fehlermeldungen	66
	Eingabe.....	67
5.4	Datentypen und Variablen.....	67
	Einfache Datentypen (Basisdatentypen)	67
	Variablen	70
	Konstanten	74

5.5	Operatoren	78
	Unäre Operatoren.....	78
	Arithmetische Operatoren	80
	Logische Operatoren	80
	Bitweise Operatoren	82
	Bedingter Operator.....	83
	Zuweisung	84
5.6	Ausdrücke.....	85
5.7	Typumwandlung	87
	Implizite Typumwandlung.....	87
	Explizite Typumwandlung (cast).....	89
5.8	Wie mache ich aus einem Ausdruck eine Anweisung?	89
5.9	Aufgaben.....	90
5.10	Zusammenfassung	91
6	Kontrollstrukturen	93
6.1	Motivation	94
6.2	{ Block }	95
6.3	Auswahl.....	97
	if-Anweisung	97
	switch-Anweisung	98
6.4	Schleifen	100
	while-Schleife	100
	for-Schleife	101
	do-while-Schleife	103
6.5	Sprünge	104
	break	104
	continue	105
	goto und Marken	106
	return.....	107

6.6	Aufgaben.....	107
6.7	Zusammenfassung.....	108
7	Funktionen und Methoden.....	109
7.1	Motivation	110
7.2	Funktionsdefinition.....	111
	Überladen von Funktionen	112
	Signatur einer Funktion	113
7.3	Funktionsdeklaration (Prototyp)	113
7.4	Rückgabewert.....	114
7.5	Aufruf	115
	Call-by-Value.....	115
	Call-by-Reference	116
7.6	inline-Funktionen	116
7.7	Rekursion	117
7.8	Lokale und globale Variablen	119
	Gültigkeitsbereich	119
	Lokale Variablen	120
	Globale Variablen	121
	Initialisierung	121
7.9	Aufgaben	122
7.10	Zusammenfassung.....	124
8	Debuggen.....	125
8.1	Motivation	126
8.2	Das Testprogramm	126
	Aufgabenstellung	126
	Programmskizze	127
	Das Programm	128

8.3	Programm in den Debugger laden	129
	Debugger starten	129
	Breakpoints setzen	131
	Variablen untersuchen.....	132
8.4	Asserts	133
	Einsatzgebiete	135
8.5	Aufgaben.....	136
8.6	Zusammenfassung	138
9	Zeiger	139
9.1	Motivation	140
9.2	Was ist ein Zeiger?.....	140
9.3	Definition.....	140
9.4	Zeiger und Adressen	141
	Adressoperator &	141
	Inhaltsoperator *	142
9.5	Zeiger auf Zeiger.....	143
9.6	Zeigerarithmetik	144
	Zuweisung	144
	Inkrementierung/Dekrementierung	145
	Addition/Subtraktion	146
9.7	Zeiger und Felder.....	147
	C-Zeichenkette.....	149
9.8	Zeiger als Funktionsargumente	149
9.9	Referenzen.....	151
	Referenzen als Parameter	151
	Referenzen als Rückgabewert.....	152
9.10	Die Parameter des Hauptprogramms	153
9.11	Aufgaben.....	155
9.12	Zusammenfassung	156

10	Strukturen und Varianten.....	157
10.1	Motivation	158
	Strukturen.....	158
	Deklaration	158
	Variablendefinition	159
	Initialisierung	159
	Kombinierte Typ- und Variablendefinition	160
	Zugriff auf Strukturen.....	160
	Erlaubte Operationen	161
10.2	Strukturfelder	161
10.3	Varianten (Unions).....	162
10.4	Strukturen und Zeiger.....	164
	Zeiger auf Strukturen.....	164
	Rekursive Strukturen.....	165
10.5	Aufgaben.....	166
10.6	Zusammenfassung.....	167
11	Klassen.....	169
11.1	Motivation	170
11.2	Begriffe aus der OO-Welt.....	170
	Was bedeutet OO?	170
	Was sind Klassen?.....	171
	Was ist ein Objekt?	171
	Was sind die Vorteile?	171
	Common Practices	172
11.3	Eine einfache Klasse.....	172
	Klasse = Struktur	173
	Date-Objekt	173
	Zugriff.....	174

11.4	Klassenmethoden	174
	Was macht eine Klasse aus?	174
	Definition der Methode (innerhalb).....	175
	Definition der Methode (außerhalb).....	176
	inline-Methode	178
11.5	this.....	179
11.6	Spezielle Klassenmethoden	180
	Konstruktor (Constructor)	180
	Copy-Konstruktor	181
	Destruktor (Destructor).....	182
11.7	Kapselung	183
	Warum Kapselung?	183
	Wie schütze ich meine Daten?	185
	Friends	186
11.8	Überladen	187
	... von Methoden.....	187
	... des Konstruktors.....	187
	... des Destruktors.....	188
	... von Operatoren	188
	... von Inkrement-/Dekrement-Operatoren	190
11.9	Dynamische Objekte.....	191
	new.....	191
	delete.....	191
11.10	static-Klassenelemente	192
	Statische Attribute.....	192
	Statische Methoden	193
11.11	Aufgaben.....	194
11.12	Zusammenfassung	194

12 Ein Ausflug in die Standardbibliothek 195

12.1	Motivation	196
12.2	Namensräume	197
12.3	Strings (Zeichenketten)	197
	Einzelne Zeichen	198
	C-Strings	199
	Die string-Klasse	200
	Ein string-Beispiel	200
	C++-string nach C-string und umgekehrt	201
12.4	Streams (Ein-/Ausgabe)	202
	Was sind „streams“?	203
	Ausgabe (ostream)	203
	Eingabe (istream)	204
	Formatierung	205
	Dateien	209
12.5	Aufgaben	213
12.6	Zusammenfassung	214

14 Vererbung215

14.1	Motivation	216
14.2	Wie erbe ich?	216
	Konstruktor	218
	Was erbe ich?	218
	Konvertierung	220
14.3	Virtuelle Methoden	220
	Abstrakte Klassen	222
14.4	Zugriff auf die Oberklasse	223
14.5	Mehrfachvererbung	224

14.6	Aufgaben.....	225
14.7	Zusammenfassung	226
15	Schablonen	227
15.1	Motivation	228
15.2	Was sind Templates (Schablonen)?	228
	Die shared_ptr-Schablone	228
	Anlegen von Shared-Pointern.....	230
	Schablone = Klasse?	230
15.3	Einsatzgebiete	231
	Ein ausführlicheres Beispiel.....	231
	Nachteil von shared_ptr	236
	Definition einer Template-Klasse	236
15.4	Weitere Templates.....	237
	auto_ptr.....	238
	vector.....	238
	list.....	239
15.5	Aufgaben.....	240
15.6	Zusammenfassung	240
16	Fehlerbehandlung	241
16.1	Motivation	242
16.2	Programmabbruch	242
16.3	Fehlercodes.....	243
	Beispiele für Fehlercodes	244
	Probleme mit Fehlercodes	245

16.4	Exceptions.....	246
	Warum Exceptions?	246
	Was sind Exceptions?	248
	Wie sieht eine Exception aus?	249
	Wie fange ich eine Exception auf?	250
	Was passiert bei einer Exception?.....	251
	Auffangen mehrerer Exceptions	253
	Weiterwerfen	254
	Vordefinierte Exception-Objekte.....	255
	Konstruktor-Exceptions.....	257
16.5	Aufgaben	257
16.6	Zusammenfassung.....	257
17	Finale	259
17.1	Motivation	260
17.2	Was man nicht braucht	260
17.3	Was noch fehlt	261
17.4	const	261
	Konstanten	261
	Zeigerkonstanten und konstante Zeiger	262
	Konstante Funktionsargumente	262
	Konstante Funktionsergebnisse	263
	Konstante Methoden.....	264
	const in der Praxis.....	264
17.5	cast	265
17.6	explicit	265
17.7	Schlüsselwörter, die keiner braucht.....	266

17.8	Variable Parameter-Liste	266
17.9	War das schon alles?	267
17.10	Die letzten Worte	267
18	Anhang	269
18.1	Lösungen der Übungsaufgaben	270
18.2	Header-Dateien.....	270
	C-Header-Dateien	270
	C++-Header-Dateien	272
18.3	C++-Bücher	274
18.4	Glossar	275
Index		279