

Inhaltsverzeichnis

1. Einleitung	9
1.1 C: eine Programmiersprache mit langer Geschichte	9
1.2 Die Anwendungsmöglichkeiten von C	11
1.3 C – Grundlage für viele weitere Programmiersprachen	12
2. Vorbereitungsmaßnahmen für das Programmieren in C	13
2.1 Texteditor und Compiler: die Grundlagen für die Programmierung in C	13
2.2 Die IDE als praktische Alternative	15
2.3 Code::Blocks für die Programmierung in C installieren	16
3. Das erste eigene Programm in C erstellen	19
3.1 Der Programmcode für das erste C-Programm	19
3.2 Das Programm in Code::Blocks ausführen	27
3.3 Die klassische Vorgehensweise zum Kompilieren und Ausführen eines C-Programms	29
3.4 Übungsaufgabe: eigene einfache Programme erstellen	34
4. Variablen und Operatoren: grundlegende Bestandteile eines C-Programms	38
4.1 Welche Aufgaben haben Variablen in der Informatik?	38
4.2 Unterschiedliche Datentypen in C	39
4.3 Variablen in unseren Programmen verwenden	41
4.4 Der Zuweisungsoperator: einer der wichtigsten Operatoren in C	46
4.5 Arithmetische Operatoren	48
4.6 Eingaben des Anwenders in einer Variablen speichern	50
4.7 Übungsaufgabe: Mit Variablen und Operatoren arbeiten	53
5. Zusammengesetzte Datentypen in C	57
5.1 Arrays: Felder für mehrere Daten des gleichen Typs	57
5.2 Mit Zeichenketten in C arbeiten	60
5.3 Struct: individuelle Strukturen für die Daten vorgeben	66
5.4 Übungsaufgabe: mit zusammengesetzten Datentypen arbeiten	69

6. Mit if-Abfragen verschiedene Optionen für die Ausführung gestalten	74
6.1 Verzweigungen erstellen	74
6.2 Vergleichsoperatoren für das Aufstellung einer Bedingung	76
6.3 Alternativen mit else einfügen	78
6.4 Das switch-Statement	81
6.5 Logische Operatoren für die Verknüpfung mehrere Bedingungen	84
6.6 Übungsaufgabe: Verzweigungen in das Programm einfügen	87
7. Schleifen und weitere Formen der Ablaufsteuerung	92
7.1 Die while-Schleife: der grundlegende Schleifentyp in C	92
7.2 Die fußgesteuerte do-while-Schleife	96
7.3 Die for-Schleife: eine weitere Alternative für sich wiederholende Programmbereiche	98
7.4 Die Schleife mit break und continue steuern	100
7.5 Mit dem goto-Statement zu anderen Programmbereichen springen	104
7.6 Übungsaufgabe: Verschiedene Schleifen im Programm verwenden	106
8. Funktionen in C	110
8.1 Was sind Funktionen und welche Vorteile bieten sie?	110
8.2 Eine erste einfache Funktion in C erstellen	111
8.3 Funktionen mit Übergabewerten	113
8.4 Funktionen mit Rückgabewerten	115
8.5 Übungsaufgabe: Eigene Funktionen erstellen	117
9. Module für eine übersichtlichere Programmstruktur erstellen	121
9.1 Was sind Module und welche Vorteile bieten sie?	121
9.2 Ein einfaches Modul mit einer weiteren C-Datei erstellen	122
9.3 Die Verwendung von Interfaces	124
9.4 Module mit Interface gestalten	125
9.5 Mehrfache Einbindungen eines Moduls verhindern	129
9.6 Übungsaufgabe: eigene Module erstellen	130
10. Die C-Standardbibliothek: vielfältige Befehle nutzen	135
10.1 Vorgefertigte Funktionen nutzen: eine erhebliche Erleichterung beim Programmieren	135
10.2 Die Dokumentation der C-Standardbibliothek	136
10.3 Mit Funktionen der Standardbibliothek arbeiten	137
10.4 Übungsaufgabe: Die Standardbibliothek verwenden	140

Inhaltsverzeichnis

11. Zeiger: ein mächtiges Werkzeug für C-Programme	144
11.1 Was ist ein Zeiger in der Informatik?	144
11.2 Die Adresse einer Variablen bestimmen	145
11.3 Zeiger im Programm verwenden	148
11.4 Übungsaufgabe: Programme mit Zeigern erstellen	153
12. Dynamische Speicherbelegung	157
12.1 Was ist dynamische Speicherbelegung und wozu dient sie?	157
12.2 Speicherplatz mit malloc() und calloc() zuweisen	160
12.3 Die Größe des zugewiesenen Speicherplatzes ändern	163
12.4 Belegten Speicherplatz wieder freigeben	165
12.5 Übungsaufgabe: Speicherplatz dynamisch vergeben	166
13. Referenzen als Übergabewerte für Funktionen	170
13.1 Übergabewerte von Funktionen: eine Kopie des eigentlichen Werts	170
13.2 Der Funktion die Adresse einer Variablen übermitteln	173
13.3 Anwendungsbeispiel: eine eigene Liste implementieren	175
13.4 Übungsaufgabe: Funktionen mit Referenzen verwenden	184
14. Fehlerbehandlung in C	190
14.1 Was sind Laufzeitfehler und weshalb müssen wir sie behandeln?	190
14.2 C: die Besonderheiten bei der Behandlung von Fehlern	192
14.3 Möglichkeiten für die Behandlung von Fehlern	193
14.4 Semantikfehler: Möglichkeiten für die Korrektur des Programms	199
15. Daten dauerhaft in Dateien abspeichern	203
15.1 Dateien: eine einfache Möglichkeit für die dauerhafte Datenspeicherung	203
15.2 Daten in eine Datei schreiben	204
15.3 Daten aus einer Datei auslesen	208
15.4 Übungsaufgabe: mit Dateien arbeiten	213
16. Datenbanken in das Programm einbinden	220
16.1 Was ist eine Datenbank und welche Vorteile bietet sie?	220
16.2 SQLite: gut geeignet für C-Programme	221
16.3 Eine Datenbank in das Programm einbinden	222
16.4 Tabellen gestalten	229
16.5 Werte einfügen, verändern oder löschen	235
16.6 Daten aus der Datenbank abfragen	238
16.7 Übungsaufgabe: mit einer Datenbank arbeiten	243

17. Grafische Benutzeroberflächen in C gestalten	249
17.1 Eine passende Technik für die Erstellung grafischer Benutzeroberflächen	249
17.2 Ein erstes Fenster erstellen.....	252
17.3 Inhalte in das Fenster einfügen.....	257
17.4 Schaltflächen für unterschiedliche Aktionen gestalten	259
17.5 Übungsaufgabe: eigene Fenster erstellen	263
18. Anwendungsbeispiel: Ein Programm für einen Immobilienmakler erstellen	267
18.1 Die Aufgaben des Programms bestimmen und Module einteilen.....	267
18.2 Neue Immobilie hinzufügen	273
18.3 Eine Vermietung oder einen Verkauf registrieren	277
18.4 Verfügbare Objekte anzeigen lassen	280
19. Ausblick	286
20. Glossar	289
21. Index	295