INFORMATIK
für Mathematiker und Physiker

Eine Einführung in C++

Skript zur Vorlesung 251-0847-00
Herbstsemester 2007, ETH Zürich[1]

Bernd Gärtner Michael Hoffmann

[1] This book originates from a set of lecture notes written by Joachim Giesen and the authors in 2005.

Contents

Contents -1

1 Introduction 5

1.1 Why learn programming? . . . 6
1.2 How to run a program . . . 11
1.2.1 Editor . . . 11
1.2.2 Compiler . . . 11
1.2.3 Computer . . . 13
1.2.4 Operating system . . . 14
1.2.5 Platform . . . 14
1.2.6 Details . . . 15

2 Foundations 17

2.1 A first C++ program . . . 18
2.1.1 Syntax and semantics . . . 19
2.1.2 Comments and layout . . . 20
2.1.3 Include directives . . . 21
2.1.4 The main function . . . 22
2.1.5 Values and effects . . . 23
2.1.6 Types and functionality . . . 23
2.1.7 Literals . . . 23
2.1.8 Variables . . . 23
2.1.9 Identifiers and names . . . 24
2.1.10 Objects . . . 24
2.1.11 Expressions . . . 25
2.1.12 Lvalues and rvalues . . . 26
2.1.13 Operators . . . 26
2.1.14 Statements . . . 29
2.1.15 Details . . . 31
2.1.16 Goals . . . 34
2.1.17 Exercises . . . 34
2.1.18 Challenges . . . 35
2.2 Integers . . . 37

2.2.1 Associativity and precedence of operators . . . 38
2.2.2 Expression trees . . . 40
2.2.3 Evaluating expressions . . . 40
2.2.4 Arithmetic operators on the type int . . . 41
2.2.5 Value range . . . 45
2.2.6 The type unsigned int . . . 46
2.2.7 Mixed expressions and conversions . . . 47
2.2.8 Binary representation . . . 48
2.2.9 Integral types . . . 50
2.2.10 Details . . . 51
2.2.11 Goals . . . 55
2.2.12 Exercises . . . 56
2.2.13 Challenges . . . 57
2.3 Booleans . . . 59
2.3.1 Boolean functions . . . 59
2.3.2 The type bool . . . 61
2.3.3 Short circuit evaluation . . . 64
2.3.4 Details . . . 64
2.3.5 Goals . . . 66
2.3.6 Exercises . . . 66
2.3.7 Challenges . . . 68
2.4 Control statements . . . 71
2.4.1 Selection: if– and if-else statements . . . 71
2.4.2 Iteration: for statements . . . 72
2.4.3 Blocks and scope . . . 76
2.4.4 Iteration: while statements . . . 81
2.4.5 Iteration: do statements . . . 83
2.4.6 Jump statements . . . 84
2.4.7 Equivalence of iteration statements . . . 85
2.4.8 Choosing the “right” iteration statements . . . 88
2.4.9 Details . . . 89
2.4.10 Goals . . . 93
2.4.11 Exercises . . . 94
2.4.12 Challenges . . . 97
2.5 Floating point numbers . . . 99
2.5.1 The types float and double . . . 100
2.5.2 Mixed expressions, conversions, and promotions . . . 102
2.5.3 Explicit conversions . . . 103
2.5.4 Value range . . . 104
2.5.5 Floating point number systems . . . 105
2.5.6 The IEEE standard 754 . . . 110
2.5.7 Computing with floating point numbers . . . 111
2.5.8 Details . . . 115

2.5.9 Goals . . . 117
2.5.10 Exercises . . . 118
2.5.11 Challenges . . . 119
2.6 Arrays and pointers . . . 123
2.6.1 Array types . . . 123
2.6.2 Initializing arrays . . . 125
2.6.3 Random access to elements . . . 125
2.6.4 Arrays are not self-describing . . . 126
2.6.5 Iteration over a container . . . 127
2.6.6 Pointer types and functionality . . . 128
2.6.7 Array-to-pointer conversion . . . 131
2.6.8 Pointer arithmetic . . . 132
2.6.9 Dynamic memory allocation . . . 135
2.6.10 Arrays of characters . . . 139
2.6.11 Multidimensional arrays . . . 143
2.6.12 Beyond arrays and pointers . . . 153
2.6.13 Details . . . 155
2.6.14 Goals . . . 156
2.6.15 Exercises . . . 157
2.6.16 Challenges . . . 160

3 Functions **163**
3.1 A first C++ function . . . 164
3.1.1 Pre- and postconditions . . . 166
3.1.2 Function definitions . . . 168
3.1.3 Function calls . . . 168
3.1.4 The type void . . . 169
3.1.5 Functions and scope . . . 170
3.1.6 Procedural programming . . . 173
3.1.7 Arrays as function arguments . . . 174
3.1.8 Modularization . . . 177
3.1.9 Using library functions . . . 183
3.1.10 Details . . . 185
3.1.11 Goals . . . 187
3.1.12 Exercises . . . 188
3.1.13 Challenges . . . 193
3.2 Recursion . . . 197
3.2.1 A warm-up . . . 197
3.2.2 The call stack . . . 198
3.2.3 Basic practice . . . 199
3.2.4 Recursion versus iteration . . . 201
3.2.5 Primitive recursion . . . 202
3.2.6 Sorting . . . 203

3.2.7 Lindenmayer systems . . . 211
3.2.8 Details . . . 217
3.2.9 Goals . . . 217
3.2.10 Exercises . . . 218
3.2.11 Challenges . . . 222

4 Compound Types **225**
4.1 Structs . . . 226
4.1.1 Struct definitions. . . . 228
4.1.2 Structs and scope . . . 230
4.1.3 Member access . . . 230
4.1.4 Initialization and assignment . . . 231
4.1.5 User-defined operators . . . 232
4.1.6 Details . . . 234
4.1.7 Goals . . . 236
4.1.8 Exercises . . . 237
4.2 Type Variants . . . 240
4.2.1 Reference types . . . 240
4.2.2 Call by value and call by reference . . . 241
4.2.3 Return by value and return by reference . . . 242
4.2.4 More user-defined operators . . . 243
4.2.5 Const-types . . . 247
4.2.6 What exactly is constant? . . . 249
4.2.7 Const-references . . . 250
4.2.8 Const-types as return types. . . . 251
4.2.9 When to use const? . . . 252
4.2.10 Goals . . . 253
4.2.11 Exercises . . . 253
4.2.12 Challenges . . . 256
4.3 Classes . . . 257
4.3.1 Encapsulation . . . 257
4.3.2 Public and private . . . 260
4.3.3 Member functions . . . 260
4.3.4 Constructors . . . 263
4.3.5 Default constructor . . . 264
4.3.6 User-defined conversions . . . 265
4.3.7 Member operators . . . 266
4.3.8 Nested types . . . 267
4.3.9 Class definitions . . . 268
4.3.10 Random numbers . . . 269
4.3.11 Details . . . 276
4.3.12 Goals . . . 277
4.3.13 Exercises . . . 277

4.3.14 Challenges 279

A C++ Operators **281**

Index **284**

B Solutions **297**