

---

---

# Contents

<b>Preface</b>		<b>xi</b>
<b>Chapter 0</b>	<b>Getting started</b>	<b>1</b>
0.1	Comments	1
0.2	#include	2
0.3	The main function	2
0.4	Curly braces	2
0.5	Using the standard library for output	3
0.6	The return statement	3
0.7	A slightly deeper look	4
0.8	Details	5
<b>Chapter 1</b>	<b>Working with strings</b>	<b>9</b>
1.1	Input	9
1.2	Framing a name	11
1.3	Details	14
<b>Chapter 2</b>	<b>Looping and counting</b>	<b>17</b>
2.1	The problem	17
2.2	Overall structure	18
2.3	Writing an unknown number of rows	18
2.4	Writing a row	22
2.5	The complete framing program	27
2.6	Counting	30
2.7	Details	31

<b>Chapter 3</b>	<b>Working with batches of data</b>	<b>35</b>
3.1	Computing student grades	35
3.2	Using medians instead of averages	41
3.3	Details	48
<b>Chapter 4</b>	<b>Organizing programs and data</b>	<b>51</b>
4.1	Organizing computations	51
4.2	Organizing data	61
4.3	Putting it all together	65
4.4	Partitioning the grading program	68
4.5	The revised grading program	70
4.6	Details	71
<b>Chapter 5</b>	<b>Using sequential containers and analyzing strings</b>	<b>75</b>
5.1	Separating students into categories	75
5.2	Iterators	79
5.3	Using iterators instead of indices	82
5.4	Rethinking our data structure for better performance	84
5.5	The <code>list</code> type	85
5.6	Taking strings apart	87
5.7	Testing our <code>split</code> function	90
5.8	Putting strings together	91
5.9	Details	96
<b>Chapter 6</b>	<b>Using library algorithms</b>	<b>101</b>
6.1	Analyzing strings	101
6.2	Comparing grading schemes	110
6.3	Classifying students, revisited	116
6.4	Algorithms, containers, and iterators	120
6.5	Details	121
<b>Chapter 7</b>	<b>Using associative containers</b>	<b>123</b>
7.1	Containers that support efficient look-up	123
7.2	Counting words	124
7.3	Generating a cross-reference table	126
7.4	Generating sentences	129
7.5	A note on performance	136
7.6	Details	137
<b>Chapter 8</b>	<b>Writing generic functions</b>	<b>139</b>
8.1	What is a generic function?	139
8.2	Data-structure independence	143
8.3	Input and output iterators	150
8.4	Using iterators for flexibility	152
8.5	Details	153

<b>Chapter 9</b>	<b>Defining new types</b>	<b>155</b>
9.1	<code>Student_info</code> revisited	155
9.2	Class types	156
9.3	Protection	160
9.4	The <code>Student_info</code> class	163
9.5	Constructors	164
9.6	Using the <code>Student_info</code> class	166
9.7	Details	167
<b>Chapter 10</b>	<b>Managing memory and low-level data structures</b>	<b>169</b>
10.1	Pointers and arrays	169
10.2	String literals revisited	176
10.3	Initializing arrays of character pointers	177
10.4	Arguments to <code>main</code>	179
10.5	Reading and writing files	180
10.6	Three kinds of memory management	182
10.7	Details	185
<b>Chapter 11</b>	<b>Defining abstract data types</b>	<b>187</b>
11.1	The <code>Vec</code> class	187
11.2	Implementing the <code>Vec</code> class	188
11.3	Copy control	195
11.4	Dynamic <code>Vecs</code>	202
11.5	Flexible memory management	203
11.6	Details	209
<b>Chapter 12</b>	<b>Making class objects act like values</b>	<b>211</b>
12.1	A simple string class	212
12.2	Automatic conversions	213
12.3	<code>Str</code> operations	214
12.4	Some conversions are hazardous	221
12.5	Conversion operators	222
12.6	Conversions and memory management	223
12.7	Details	225
<b>Chapter 13</b>	<b>Using inheritance and dynamic binding</b>	<b>227</b>
13.1	Inheritance	227
13.2	Polymorphism and <code>virtual</code> functions	232
13.3	Using inheritance to solve our problem	237
13.4	A simple handle class	243
13.5	Using the handle class	247
13.6	Subtleties	248
13.7	Details	250

<b>Chapter 14</b>	<b>Managing memory (almost) automatically</b>	<b>253</b>
14.1	Handles that copy their objects	254
14.2	Reference-counted handles	260
14.3	Handles that let you decide when to share data	263
14.4	An improvement on controllable handles	264
14.5	Details	268
<b>Chapter 15</b>	<b>Revisiting character pictures</b>	<b>269</b>
15.1	Design	269
15.2	Implementation	278
15.3	Details	288
<b>Chapter 16</b>	<b>Where do we go from here?</b>	<b>291</b>
16.1	Use the abstractions you have	291
16.2	Learn more	293
<b>Appendix A</b>	<b>Language details</b>	<b>295</b>
A.1	Declarations	295
A.2	Types	299
A.3	Expressions	305
A.4	Statements	308
<b>Appendix B</b>	<b>Library summary</b>	<b>311</b>
B.1	Input-output	311
B.2	Containers and iterators	314
B.3	Algorithms	322
<b>Index</b>		<b>325</b>