

D. Index

#define.....	365	composition.....	218, 219, 368
#ifdef.....	365	conditional expression.....	7, 362, 367
#ifndef.....	365	const.....	4, 146
#include.....	87	const_cast.....	5
::.....	<i>See</i> scope resolution operator	constructor.....	151, 152, 241, 368
->.....	77	copy.....	154
abstract class.....	269, 365	default.....	152
abstracting.....	225	non-default.....	153
access modifier.....	117, 136, 365	container.....	341, 368
accessor.....	144	continuous data.....	121
address-of operator.....	365	coupling.....	86, 115, 367
aggregate data-type.....	71	cout.....	9, 359, 368
aggregation.....	219	cstdlib.....	22
argc.....	362	cstring.....	22
argv.....	362	data.....	120
array.....	17, 360, 366	data abstraction.....	115
arrow operator.....	366	default parameters.....	103, 368
assert.....	55, 359, 366	delete.....	7, 362, 370
association.....	219	dependency.....	219
base class.....	237, 366	dereference operator.....	370
binding.....	229, 258, 366	derivation.....	218, 370
bitwise operator.....	366	derived class.....	237, 370
bool.....	4, 366	design document.....	370
Boolean operators.....	366	Design Document	
callback.....	289, 290, 367	Algorithms.....	33, 40
callee.....	290, 367	Data Structure.....	42
caller.....	290, 367	Data Structures.....	33
cast.....	5	Design Overview.....	33, 36
casting.....	367	Error Handling.....	33, 44, 48
catch.....	65	File Format.....	33, 43
catch-all.....	65	Interface Design.....	33, 37
cctype.....	22	Problem Description.....	33, 35
char.....	4, 367	Structure Chart.....	33, 38
child class.....	237	destructor.....	151, 156, 370
chmod.....	90	discrete data.....	121
cin.....	10, 359	double.....	4, 370
cin.clear().....	54	downcasting.....	261, 371
cin.fail().....	53	dynamic_cast.....	5
cin.ignore().....	54	early binding.....	229, 371
class.....	115, 135, 367	EID.....	63, 371
class template.....	310, 368	encapsulation.....	114
client.....	98, 368	eof.....	371
cohesion.....	72, 86, 115, 367	error flags.....	371
functions.....	14	error handling.....	371
coincidental.....	15	escape sequences.....	371
comments.....	355, 356, 367	expression.....	371
communicational.....	14	extraction operator.....	7, 176, 373

file i/o	43, 50
flags	62
float	4
friend	192, 193, 194, 373
fstream	10, 373
function pointer	107, 128, 230, 373
function signature	99, 107, 305, 373
function template	302, 303, 373
functional	14
g++	89, 90
generic algorithms	281, 302
generic data-structures	281, 310
getline	10, 373
getter	122, <i>See</i> accessor
has-a	218, 219
header file	105, 137, 145, 373
if	359
ifstream	360, 374
implementation	227
implementation file	87
infix	174
information hiding	115
inheritance	218, 374
inheritance indicators	250
inline	105, 145, 374
insertion operator	7, 175, 374
instantiate	136
int	4, 374
interface	227
interval data	121
invariant	222, 374
iomanip	9, 376
iostream	9, 376
is-a	218, 220
iterator	331, 376
iterators	330
iternator	374
late binding	229, 376
linked list	317, 376
list	345
local variable	376
logical	15
long	4
long double	4
loop	
do-while	13, 360, 370
for	13, 360, 373
while	12, 360, 384
makefile	91, 137, 376
map	347, 348, 349
member function	115, 117
member variable	115, 376
method	115, 117
modularization	115
modulus	8, 376
multi-dimensional arrays	17, 362, 377
mutator	144, 377
new	7, 362, 371, 377
node	317, 377
null	377
NULL	377
object	115, 377
object file	90, 377
ofstream	360, 378
operator	99, 173, 201, 378
-- decrement	181, 182, 186, 188, 196, 202, 212
- negative	183, 186, 188, 196, 202
- subtraction	177, 186, 188, 196, 202, 212
! not	183, 186, 188, 196, 202
!= not equals	184, 186, 188, 196, 202, 212
% modulus	177, 186, 188, 196, 202
&& and	207, 213
() function call	206, 212
* multiplication	177, 186, 188, 196, 202
*= multiply by	179, 186, 188, 196, 202
/ division	177, 186, 188, 196, 202
/= divide by	179, 186, 188, 196, 202
[] square bracket	205, 213
or	207, 213
+ addition	177, 186, 188, 196, 202, 208-212
++ increment	181-188, 196, 202, 209, 211-212
+= add onto	179, 186-196, 202, 208, 211-213
< less than	184, 186, 188, 196, 202, 212
<< insertion	175, 186-196, 202, 212, 243, 253
<= less than or equal	184-188, 196, 202, 212
= assignment	204, 212, 213
-= subtract from	179, 186, 188, 196, 202, 212
== equivalence	184-196, 202, 209, 212
> greater than	184, 186, 188, 196, 202, 212
>= greater than or equal	184-188, 196, 202, 212
>> extraction	176-188, 196, 202, 212, 243, 253
parent class	237
pass-by-pointer	378
pass-by-reference	16, 378
pass-by-value	16, 378
pointer	378
arithmetic	20
declare	20
polymorphism	378
private	117, 136, 378
procedural	14
properties	120
protected	248, 380
prototype	103, 380
pseudocode	40, 380
public	117, 136, 380
pure virtual function	269, 270
random access	318

redefining	242	styleChecker	383
reinterpret_cast	5	switch	362
scope	380	tabs	383
scope resolution operator	138, 380	TAR	92, 383
separate compilation	380	template	281
sequential	14	template prefix	303, 311
setf()	359	temporal	14
setter	122, <i>See</i> mutator	this	140, 383
setw()	9	throw	64, 383
short	4	throw list	383
sizeof()	7, 380	try	64, 383
slicing problem	261, 262, 381	type parameter	303, 311
source file	87, 137	UML	384
standard template library	341	UML class diagram	42, 72-74, -128-165, 219-257
static	164, 165, 381	universalizing	225, 384
static_cast	5	unsigned	4
STL	381, <i>See</i> standard template library	upcasting	262, 384
strings	23, 360, 381	variable	384
c-strings	368	variant	222, 384
struct	74	vector	23, 342, 343, 344
structure	381	virtual function	258, 384
structure chart	38, 381	void pointer	289, 295, 384
structure tag	381	v-table	127, 129, 230, 384

This project would not have been possible without the keen eye, perspective, and attention to detail of Adam Harris. He produced many of the illustrations, drafted many of the chapters, and proofed every single word.

I would also like to thank Scott Burton and Scott Ercanbrack for their contribution to this effort.