

[Orders](#)

[Our Blog](#)

[Excerpts](#)

[Savings](#)



Dorset House Publishing

Book Titles . . .

Authors . . .



Web Dorset

Google

[Savings](#)

[Orders](#)

[Find a Store](#)

[Excerpts](#)

[Titles](#)

[Subjects](#)

[Authors](#)

[About Us](#)

[Proposals](#)

[Review Us](#)

[Contact Us](#)



Index of

What Every Programmer Should Know About Object-Oriented Design

by [Meilir Page-Jones](#)

foreword by Larry L. Constantine

ISBN: 978-0-932633-31-6

©1995 392 pages hardcover



Rate this Book.

\$44.95 (plus shipping)

Subject(s): [Object-Oriented Design](#)

Add to Cart

Payments by **PayPal**

***For UPS Ground within U.S. only.**

For more info., or for Int.'l or rush orders, [click here.](#)

View Cart

Order from an Online Bookstore

Find Out How to Save 50%—A Website Exclusive!

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#)

[N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [V](#) | [W](#) | [Y](#) | [Z](#)

A

abstract data-type *xix*, 14, 53, 80, 339

class vs. 28-29, 178, 212, 294, 308

accessor method 78-79, 81-84, 85, 92, 94, 95, 238, 239, 339

account number example 185-86

action message 145, 339

actual argument 79, 114

Ada 53, 88, 89, 120

ADT *See* abstract data-type aggregate object 72, 100-105, 116, 260-62, 328, 339

aircraft example 112-14

airplane example 260-62

alternate cohesion 303, 308, 335, 339

analysis *See* requirements analysis

ancestor class 339

AOP *See* apex of polymorphism

apex of polymorphism 270-71, 273, 277, 281, 340

application domain 200-201, 219, 340

architectural domain 200-201, 216, 219, 325, 332, 340

architecture 340

architecture model 340

argument 19-21, 44, 141, 145-48, 329, 340

formal 41, 114

input and output 9, 19, 20, 46, 84

notation 113-14, 136, 137, 160

role in message 18, 19-21, 145

aspect 278, 340

assertions 62, 63, 66, 149, 150

assignment 7, 15

asterisk symbol 83

asynchronous messaging 72, 111, 121-34, 135, 165, 328, 340

Atkinson, C. 122, 131, 357

attribute 83ff., 340

awkward behavior 298-99, 308, 329, 335, 340

[top](#)

B

balanced tree example 39-42, 43, 78

bank customer example 220, 223-24

base class 340

BCPL 53

behavior 44, 225, 226-27, 231-32, 239, 240, 250, 253, 334, 335, 340

See also awkward, dangerous, ideal, illegal, incomplete, and irrelevant behavior; principle of closed behavior

in class interface 296-302, 307-8

of a subclass 231-32

Bertino, E. 322, 357

Blitz Guide 337-38

Booch, G. 122, 357

book chapter example 109-10

broadcast message 122, 129-30, 135, 328, 340

brokerage example 300-301

business domain 200-201, 203-4, 205, 206, 216, 333, 340

[top](#)

C

C 48, 53-54, 89

C++ *xviii*, 15, 22, 46, 48, 50, 53, 81, 84, 85, 87, 88, 91, 120, 136, 137-38, 193, 196, 211, 273, 276, 332, 337, 338

callback 124-25, 133-34, 340

caller 13

CASE 55, 57, 73, 77, 192, 199

chair example 97, 102, 103

Chambers' Twentieth Century Dictionary 357

chemical factory tank example 148-49

chessboard example 226-27, 232

Chidamber, S. 358

Christerson, M. 359

circle example 239

class *xix*, 2, 3, 24-29, 44, 46, 47, 50, 52, 59, 68, 69, 93, 94-95, 100, 308, 325-30, 332-35, 338, 341

- abstract data-type vs. 28-29, 178, 212, 239, 308, 334

- aspect 278

- as biological organ 180

- domains of 200-207

- instance vs. 279

- instantiation 24, 25

object vs. 25, 173, 174

properties 225ff., 353

quality 285, 293-306, 308

symbol 78, 92

class based 47, 341

class cohesion 178, 200, 211, 212-18, 219, 283, 285, 293, 307, 329, 341

See also ideal class, mixed-domain, mixed-instance, and mixed-role cohesion

class coupling 182, 208, 341

class-external-interface diagram 72, 79-81, 90, 92, 93, 96, 136, 341

See also pin-out diagram

class-inheritance diagram 72, 97-100, 107, 108, 341

single-inheritance and 98-99

class-inheritance hierarchy 44, 46, 49-50, 72, 115, 135, 157, 247, 262-63, 279, 280, 286, 327, 341

class interface 285ff., 294, 329, 341

behavior support in 296-302, 329

method cohesion in 302-6

quality of 293-306

state-space support in 294-96

class-internal-design diagram 72, 89-91, 92, 136, 138, 341

types of arrows 90, 92

class invariant 54, 62, 63, 66, 72, 153, 178, 225, 233-35, 238, 239, 247, 250, 251, 252-53, 254-55, 257, 296, 326, 327, 334, 341

class library *See* object-oriented libraries

class message 27, 46, 49, 341

class method 27, 28, 44, 46, 49, 86, 95, 338, 341

class migration 161-62, 170, 251, 258, 322, 334, 341

class-reference set:

 direct 207-8, 209, 211, 219, 343

 indirect 207, 208, 209, 210, 219, 221, 347

class variable 27, 28, 44, 46, 264-65, 338, 341

client-server system example 163-64

cloned code 41, 42, 43, 45

closed behavior *See* principle of closed behavior

COBOL 89

code 55, 62

See also object-oriented code

 key to 7

Cohen, S. 361

cohesion 52, 60, 69, 177, 181, 198, 341

See also class cohesion

 in class interface 303-6, 308

Coleman, D. 358

component object 72, 104, 105, 106, 107, 116, 117, 342

concurrency 72, 95, 121-34, 135, 328, 342

See also method-level, object-level, and
 system-level concurrency

conditional [state] transition 145, 146, 147, 151, 152,
153, 342

cone of definition 283

cone of polymorphism (COP) 270-71, 273, 274, 277, 279, 281, 284, 328, 342

confinement of behavior 232, 250, 342

confinement of state-space 229, 231, 250, 342

connascence 52, 177, 179, 183-95, 196, 197, 198-99, 212, 291, 297, 301, 319, 326, 328, 332ff., 342

 abuses of 192-95

 maintainability and 190-92

 varieties of 183-88, 196

constant 342

Constantine, L. *vii, xv-xvi*, 52, 69, 71, 75, 76, 77, 182, 197, 198, 302, 358, 361, 362

construction 62-63, 342

container class 43, 45, 103, 106, 202

continuous variable 148-49, 151, 153-54, 342

contranascence 188-89, 190, 196, 199, 326, 332, 342

Cook, S. 144, 358

COP *See* cone of polymorphism

core representational variable 288-89, 293, 326, 334, 335, 342

correctness 62

coupling 52, 60, 69, 98, 177, 181-82, 198, 319, 342

Cox, B. 59, 358

creation 342

cuboid example 82-83

customer information system exercise 171-72, 173-74

D

daemon 89

Dahl, O.-J. 52, 358

dangerous behavior 297, 307, 335, 342

Daniels, J. 144, 358

data *xix*, 10, 44, 160, 244

- analysis 61

- object orientation and 20, 22, 44, 46, 48, 61, 160

- symbol 80, 85, 96, 114, 137, 160

database access 72, 155-62, 170

data-type 14, 48-49, 80, 202, 339

deferred class 87-88, 94, 281, 338, 343

deferred method 86, 87-88, 95, 278, 281, 343

degrees of freedom 254-55, 334, 343

delegation 269, 343

DeMarco, T. 358

derived class 343

descendant class 343

design *xviii*, 55, 59-60, 61-62, 66, 177, 178, 191, 325, 330, 343

- by contract 54, 178, 236, 250

- criteria 52, 177, 181-83, 283

- notation 72, 73, 75, 76

- owner's manual 331-35

Dijkstra, E. 52, 358

dimensionality 252, 254-56, 343

dimensions 227-29, 252, 343

direct class-reference set *See* class-reference set

disnascence 188, 196, 343

documentation 329, 330

dog owner example 217-18, 312-14

domain 200-207, 211, 216, 219, 267, 326, 333, 343

See also application, architectural,
business, and foundation domains;
fundamental subdomain

sources of 205-7, 211, 325

double colon *See* qualifier symbol

dynamic binding 37, 45, 115-16, 343

dynamic classification 343

dynamic connascence 187-88, 196, 343

[top](#)

E

Edwards, J. 71, 144, 359

effective method 344

Eiffel *xvii*, 15, 32, 48, 54, 79, 81, 84, 85, 88, 91, 120, 189,
211, 273, 277, 278, 279, 282, 334, 338

Einstein, A. 55, 61

Embley, D. 358

encapsulation 2, 3, 9-12, 14, 44, 46, 47, 52, 58, 61, 70,
102, 177, 179-83, 189-90, 191-92, 193, 194-95, 196, 197,
198, 212, 225, 289, 291, 326, 332, 335, 344

levels 180-83, 190-91, 196, 244, 348

testing and 193

encumbrance 177, 200, 207-12, 215, 216, 219, 220, 221,
267, 293, 307, 326, 333, 344

of fundamental subdomain 208, 209

essential event-type 343

essential model 162, 343

event-activity manager 344

event model 344

event-stimulus recognizer 344

event thread 344

exception 63, 66, 12-21, 344

exception detector 120, 135, 344

exception handler 63, 120, 344

exception indicator 120-21, 126, 137, 162

explicit connascence 184, 192, 196, 198, 344

extensibility 63, 65, 285, 330, 344

extension of behavior 232, 345

extension of state-space 230-31, 345

external dimensionality 256, 345

external state 345

extrinsic class 214-15, 217, 345

[top](#)

F

Fad of the Year 60

failures 62-63

fan-in 91

fan-out 181, 182, 319, 345

feature 345

Firesmith, D. 359

first-order design 244, 345

formal argument 79-80, 114, 345

formal interface 79, 80

formal signature 79, 327

foundation domain 200, 201-2, 203, 205-6, 209, 219, 220, 221, 325, 332, 345

Freedman, D. 325, 359

friend function 50, 136, 137-38, 193, 196, 332, 345

frozen method 86, 87, 345

functional cohesion 302-3, 305-6, 308, 345

function-style method 81-84, 85, 92, 95, 96, 120, 345

fundamental subdomain 201-2, 208, 209-10, 211, 216

[top](#)

G

garbage collector 17, 156, 318, 346

generic class 41, 45, 78, 94, 103-5, 117, 275-78, 279, 328, 334

genericity 2, 3, 39-43, 44, 45, 53, 338, 346

perils 43, 275-78

Genghis the Perverse 298, 304

Gill, S. 362

glider example 29-34, 46, 49, 101-2, 262

Goldberg, A. 52, 359

Gordian hammock 61

Graham, I. 359

graphical user interface (GUI) *xviii*, 50, 64, 66, 175, 346

gray boxes 101, 133

GUI *See* graphical user interface

[top](#)

H

hacking 335, 346

handle 14-15, 16, 17, 25, 26, 44, 46, 48, 49, 102, 118, 156, 346

 physical memory address 16

 role in message 18, 19, 20-21, 129

Harel, D. 144, 359

hash table example 42-43, 186, 190

Hatley, D. 164, 165, 359

Haydn, F. 197

Henderson-Sellers, B. 71, 144, 359

Holland, I. 211, 360

hominoid example 4-10, 12-20, 23-25, 27-28, 46, 78-79, 90, 93, 96, 136, 137-38, 220, 221-23, 300

homomorphism 63

horizontal partitioning 174, 346

human interface 155, 166-69, 170, 330, 346

hybridity 85

hybrid system 77, 114

[top](#)

I

Ichbiah, J. 53

ideal behavior 301-2, 308, 346

ideal class cohesion 218, 219, 223, 308, 332, 333, 346

ideal method cohesion 305, 308, 346

ideal states 296, 307, 346

illegal behavior 296-97, 307, 328, 346

illegal states 294, 307, 335, 346

immutable class 94, 346

immutable object 48, 231, 347

imperative message 22, 23, 24, 79, 306, 347

implementation hiding 2, 3, 11-13, 14, 37, 44, 45, 119, 291, 347

implicit connascence 184, 192, 196, 197, 198, 199, 332, 347

inappropriate states 295, 307, 328, 335, 347

incomplete behavior 298, 307-8, 328, 335, 347

incomplete states 295, 347

indirect class-reference set *See* class-reference set

information hiding 2, 3, 11-13, 14, 44, 53, 347

informative message 22, 23, 79, 306, 319, 320, 328, 347

inheritance *xix*, 2, 3, 29-34, 44, 47, 61, 63, 70, 97-100, 178, 222, 225, 239-40, 259, 282, 294, 327, 328, 332ff., 347

abuses *xv*, 193-94, 196, 260-69, 279

encumbrance and 220, 221

is a test 32, 35, 46, 265-69, 289, 327

maintenance and 29

of a method 86-87, 95, 283

multiple 33-34, 97-100, 106, 108, 188-89,
261-62, 263-65, 266-67, 281, 283-84, 287, 289,
315, 327, 350

notation 30, 87, 92, 97-98, 99, 100, 106, 137

privacy vs. 86

text vs. diagrams 98-99, 106

inheritance hierarchy *See* class-inheritance hierarchy

instance 31, 32, 347

class vs. 279

object vs. 32

instance constant 118, 347

instance message 347

instance method 27, 32, 44, 46, 91, 95, 338, 347

instance variable 27, 32, 44, 46, 49, 264, 265, 338, 347

instantiation 24, 25, 27, 89, 120, 141, 233, 338, 348

from deferred class 87, 88

from literal class 49, 87

from mix-in class 287, 289

integrated circuit (IC) 59, 66, 68, 69, 80-81

internal dimensionality 256, 348

internal state 348

interrogative message 22, 23, 79, 306, 320, 328, 348

intrinsic class 215, 216, 348

invariant *See* class invariant

irrelevant behavior 298, 307, 335, 348

is a test 32, 35, 46, 265-69, 289, 327

iterated message 116-17, 129, 328, 348

iterator symbol 117, 135, 348

[top](#)

J

Jacobsen, I. 359

Jonsson, P. 359

[top](#)

K

Kant, I. 177

Kay, A. 52, 359

Keats, J. 71

Kemerer, C. 358

Kuhn, T. 56, 360

Kurtz, B. 343

[top](#)

L

LaLonde, W. 238, 360

Land of the Midnight Fix 194

Law of Demeter 211-12, 219, 333, 348

Lieberherr, K. 211, 360

Liskov, B. 14, 53, 238, 360

literal class 48-49, 87, 94, 348

literal object 48, 348

little hat notation 99, 106, 138-39, 222

living anthropoid example 49-50

locking 130-31, 348

logically persistent object 156, 348

logically volatile object 349

Love, T. 360

[top](#)

M

machine example 143-45, 147-48, 153

maintainability 29, 61, 63-64, 65, 66, 189, 190-92, 194, 197, 199, 212, 285, 328, 330

management *See* object-oriented management

manager example 241-47, 252, 256

Martin, J. 323, 360

Martino, L. 322, 357

McConnell, S. 360

Mealy, G. 142, 149, 151, 360

Mellor, S. 140, 153, 165, 361, 362

memory 9, 26-29, 44, 50, 120

Mental Health Warning 240

message 2, 3, 18-24, 44, 46, 49, 111, 113, 328, 338, 349

arguments 19-21, 145-48

arrow notation 112, 118, 123, 135, 165

objects in 18-19

polymorphism and 38, 269-75

self in 118-19, 291

SOP and 273-75, 334

structure 18-19

traditional techniques vs. 19, 140

types of 22-24, 306

message forwarding 268-69, 279, 282, 316, 327, 328, 334, 349

message queue 125, 128-29, 135, 349

parallel 126

priority 126, 128-29, 135

method *xix*, 10, 11, 26, 29, 78-79, 81-88, 93, 95, 159, 239, 279, 338, 349

cohesion 302-6, 308

name 19, 306

notation 78-91, 92, 93, 94, 120

method-level concurrency 95, 127, 138, 349

method postcondition 54, 72, 117, 178, 225, 236, 237, 238, 239, 241, 243, 246, 250, 326, 349, 352

method precondition 54, 72, 117, 178, 225, 236, 237, 238, 239, 241, 242, 245, 250, 326, 349, 352

method ring 119, 289-93, 307, 328, 335, 349, 354

Meyer, B. *vii*, 34, 54, 79, 189, 236, 238, 283, 360

mixed-domain cohesion 4, 213, 214-17, 218, 219, 220, 221, 223, 326, 332, 333, 334, 349

mixed-instance cohesion 213-14, 217, 218, 219, 220, 223, 326, 332, 333, 349

mixed language 22

mixed-role cohesion 213, 217-18, 219, 220, 223, 287, 320, 326, 332, 333, 349

mix-in class 278, 285, 286-89, 307, 313, 321, 327, 333, 349

model 349

modifier method 78-79, 81-84, 85, 92, 94, 95, 350

Module of the Month Club 180

Moore, E. 142, 149, 151, 360

MOSES *xvi*, 71

multiple classification 350

multiple cohesion 303-5, 308, 335

multiple inheritance 33-34, 97-100, 106, 108, 188-89, 261-62, 263-65, 266-67, 281, 283-84, 287, 289, 315, 327, 350

multiple messages 122, 125-29, 141, 297, 298, 335

mu-meson 51

mutable class 94, 350

mutable object 350

[top](#)

N

navigational query 157-60, 350

nondirectional connascence 185

non-inheritable method 86, 350

non-overridable method 86

Nygaard, K. 52, 358

O

object *xix*, 1, 10, 21-22, 25, 46, 50, 69, 78, 102, 225, 244, 338, 350

- as black box 12-13

- class vs. 25, 173, 174

- class migration 161-62

- data vs. 20-21, 22, 46, 61, 160

- instance vs. 20-21

- memory requirements 26-29, 50

- object-level concurrency and 128

- persistence 50, 156-57, 170, 171, 173

- replication 174

- roles 21-22

- state 228

- structure 11, 24-26, 44

object-action paradigm 166, 350

object-aggregation diagram 72, 97, 100-105, 106, 107, 109-10, 112, 260-62, 350

- cardinality notation 103, 105, 106

- encapsulation violation 102

- notation of arrows 104, 106, 109, 112, 135

- orientation on page 104

object based 47, 89, 350

object-communication diagram 72, 91, 92, 111ff., 135, 137, 139, 350

graphical tips 114

physical boundaries on 165, 170

polymorphism in 115

object identifier 15, 350

See also handle

object identity 2, 3, 14-17, 44, 47, 350

object-interaction/timing diagram 72, 122, 131-34, 135, 350-51

object-level concurrency 95, 127, 128, 133-34, 136, 138, 351

object module 78, 351

object orientation 44, 51ff., 60-65, 155-56, 190, 205, 206, 282, 328

as engineering discipline 59-60, 69

organizational changes and 65

researchers 52-54

social context 55-58

object oriented 1-2, 3ff., 19, 351

code 4, 5, 6, 7, 8, 29-30, 33, 37, 39, 62, 63, 71, 76, 112, 159, 177, 251, 334

design *xviii*, *xx*, 59-60, 61-62, 66, 75ff., 178, 192-95, 218, 226, 240, 256, 324, 325

environment 2, 20, 22, 26, 37, 44, 46, 47, 63, 69, 121, 188

evolutionists 56ff., 66, 68

gurus 2, 9

languages 63, 64, 69, 76-77, 79, 92, 161, 189, 202, 322, 326

libraries *xviii*, 64, 65, 77, 97, 184, 189, 193-94, 205, 220, 221, 238, 254, 320, 326, 329, 332ff., 341

management 65, 67

program 50

programming *xvii*, xx, 55, 177

properties 2, 3, 44, 68

reactionaries 56, 57, 60, 66, 68, 69

revolutionaries 55-56, 66, 68, 69

software *xviii-xix*, 59-60

systems *xvi*, *xix*, 71, 77, 111, 121, 155, 160, 181, 188, 190, 192-95, 200

traditional techniques vs. 20, 29, 58, 66, 89, 155, 177, 309, 319, 334

Object-Oriented Database-Management System (ODBMS) 50, 55, 70, 128, 156, 157-58, 161

Object-Oriented Design Notation (OODN) *xvi*, 71ff., 351

analysis notation vs. 73

goals 75-77

object-query language (OQL) 70

object structured 47, 351

Odell, J. *vii*, 323, 360

OID *See handle ontogeny recapitulates phylogeny example* 54

OODL *xvi*

OODN *See* Object-Oriented Design Notation
optimistic concurrency control 130-31, 351

Ortega y Gasset, J. 1

Overgaard, G. 359

overloading 38-39, 45, 84-85, 92, 319, 338, 351

overriding 38, 44, 86-87, 88, 95, 241, 248, 251, 256, 327, 334, 338, 351

notation 87, 88, 92, 137

[top](#)

P

package 53, 75, 88-89, 92, 94, 207, 351

Page-Jones, M. *xv*, 14, 60, 71, 77, 182, 361

panda example 263-65, 280, 282-83

parameterized class 351

Parnas, D. 53, 361

partitioning 171, 173-74, 330

Pauli, W. 51

Pavarotti example 257

perfect technology 351-52

persistence 50, 156-57, 170, 171, 173

logically persistent object 156, 348

physically persistent object 156, 352

symbol 156, 160

persistent object 352

pessimistic concurrency control 130-31, 352

pin-out diagram 72, 80-81, 84, 85, 91, 92, 93, 114, 136, 352

See also class-external-interface diagram

Pirbhai, I. 164, 165, 359

pointer 15, 137, 352

polygon example 34-38, 87-88, 98, 115-16, 247-49, 253, 258, 284

polymorphism *xix*, 2, 3, 34-39, 42, 44, 45, 64, 70, 115, 130, 135, 166, 178, 245, 249, 259, 268, 279, 328, 352

 danger of 115, 259, 269-78

 genericity and 275-78

 in messages 269-75, 276

 of target object 115

Porter, H. 361

postcondition *See* method postcondition

precondition *See* method precondition

principle of closed behavior 247-49, 250-51, 253, 256-57, 327, 334, 352

principle of contravariance 241, 243, 250, 327, 333, 352

principle of covariance 241, 244, 250, 327, 333, 352

principle of type conformance 178, 225, 238-40, 244, 246-47, 282, 289, 293, 315, 316, 327, 333, 352

 constraints 246-47

private method 11, 91, 95, 327, 338, 353

private variable 11, 83, 99, 130, 239, 327, 335, 338, 353

procedural modules 53, 88, 180, 335

procedure 10, 13, 181, 206

procedure-style method 81, 84, 92, 95, 353

processor-interconnect diagram 72, 165-66, 170, 353

Professor Rossini example 39

programmers *xviii*, *xix*, 57, 65, 76, 120, 330

two kinds *xvii*, *xviii*

programming 55, 65

See also object-oriented programming

programming language *xvii-xviii*, 2, 3

project leader name example 158-59

properties 225ff., 353

protected method 338, 353

protected variable 338, 353

public method 11, 95, 99, 338, 353

public variable 11, 338, 353

Pugh, J. 238, 360

[top](#)

Q

qualifier symbol 114, 353

quality of a class interface 293-306

quality vector 353

query language 70

[top](#)

R

Rabi, I. 51

real-time systems 24, 61, 66, 111, 130, 132, 151, 152,
153-54, 187

rectangle example 237-38, 240, 252-53, 254-56, 257-58,
286-95, 297, 303-5, 309, 315, 316-19

redefinition 353

relational database 50, 155, 170

relational DBMS 70, 160

reliability 62, 63, 64, 66, 285, 353

repeated inheritance 353

replicated behavior 300-301, 308, 309, 329, 335, 353

requirements analysis *xviii*, 61, 65, 66, 73, 206, 324, 329, 330

rescue clause 63, 120, 353

reusability 43, 55, 62, 63, 65, 66, 67, 177, 180, 202, 204, 205-7, 211, 216, 217, 218, 219, 285, 307, 320, 321, 326, 328, 330, 332, 333, 353

Richards, M. 53, 361

ring of methods *See* method ring

road-vehicle example 229-31, 232

Robson, D. 52, 359

robustness 62-63, 64, 65, 66, 120, 285, 354

room example 265-69, 283, 309, 315-16

Ross, D. 361

Ruble, D. 168

[top](#)

S

sales commission example 213-14

sales system example 167-69

scanning query 157, 160, 354

scope of polymorphism (SOP):

of a deferred class 284

generic classes and 276-78

message and 273-75

of a method 270-72, 273, 274-75, 279, 280, 281, 284, 328, 334, 354

mnemonic verse 275, 277

ragged 271-72, 281, 284, 334, 328

of a variable 272-75, 279, 281, 328, 334, 354

scope-resolution symbol *See* qualifier symbol

second-order design *xix*, 244, 354

self 48, 91, 118-19, 137, 291, 338, 354

 first usage 118

 second usage 119

Self 77

semicolon 7, 20, 80, 84

seminar company example 145-47

sender object 18, 22, 37, 111, 129-30, 338, 354

Sharble, R. 361

shipment unit example 116-17, 309-11, 319-20

Shlaer, S. 153, 361

signature 19-20, 39, 46, 84, 92, 112, 239, 338, 354

Simula 52

single inheritance 33-34, 354

Skolnik, M. 59, 362

Smalltalk *xviii*, 15, 20, 22, 48, 52, 85, 91, 211, 273, 338

software development 2, 51, 55, 56, 57, 62-63, 66, 67, 68

manufacturing and 57-58

SOP *See* scope of polymorphism

sorted tree example 276-78

stack example 236, 280, 282, 295, 296-97, 302

standard method 354

state 10, 25, 151, 228, 335, 354

 in class interface 294-96, 307

 definitions 149-50

 nested 140, 142-45, 150, 151

 variables 149

state retention 2, 3, 13-14, 44, 47, 354

state-space 72, 149, 178, 225, 226-31, 232, 233-35, 248, 250, 254-56, 354

 of a subclass 229-31

state transition 141, 149, 151, 152, 162, 228, 354

state-transition diagram 72, 140-54, 162, 168, 354

static binding 115, 354-55

static connascence 185-87, 355

Stroustrup, B. 53-54, 362

structured analysis 58

structured design *xv, xvi*, 14, 58, 60, 69, 89, 91, 177, 181, 186, 190, 197, 198, 212, 244, 302, 304, 305, 309, 319

Structured Design Notation (SDN) 77, 78, 112, 113, 115, 123, 158, 159, 160, 170, 355

structured programming 181-82

structured-query language (SQL) 70

structured techniques 61, 66, 71

subclass 29, 239, 327, 338, 355

 as subtype 225, 239, 240-47, 333

subdomains 201ff.

subject-oriented systems 58

subroutine 9, 10, 19, 62, 113, 179, 180, 196

subtype 225, 238, 239-47, 333, 355

superclass 29, 334, 338

synchronous messaging 111-21, 122, 135, 355

Synthesis *xvi*, 355

system architecture 155, 162-66, 170

system-level concurrency 126-27

system size 57

[top](#)

T

target method 355

target object 18, 19, 22, 36, 37, 38, 111, 112, 115, 125ff.,
135, 338

task-interconnect diagram 166

technology-interconnect diagram 72, 164, 170, 355

text 72, 92

 for inheritance hierarchies 98-99, 106

 for message signature 114, 166

triangle example 233-35, 252, 257

trigger message 145, 355

two-phase commit 131, 355

type *See* abstract data-type

[top](#)

U

Uniform Object Notation (UON) *xvi*, 71, 75, 76, 102, 355

[top](#)

V

variable *xix*, 7, 10, 26, 29, 100, 279

 name 102

 public vs. private 11

 state-transition diagrams and 141, 153

vertical partitioning 174, 355-56

video-rental store example 188-89, 283

volatile object 156, 158, 356

[top](#)

W

wait-point symbol 132, 133

walkthroughs 62, 256, 263, 299, 325-30

walled city example 11

Ward, P. 140, 165, 362

Weasel example 194-95

Webster's Third New International Dictionary 362

Wegner, P. 47, 69, 362

Weinberg, G. 325, 359

Weiss, S. *vii*, 71, 75, 361

Wiener, R. 54, 236, 362

Wilkes, M. 9, 362

Wheeler, D. 362

Whitby-Stevens, C. 53, 361

window-layout diagram 72, 167-68, 169, 170, 174-75, 356

window-navigation diagram 72, 167, 168-69, 170, 171-72, 175, 330, 356

Woodfield, S. 358

Wordsworth, W. *xvii*, 275

[top](#)

Y

Yourdon, E. *vii*, *xv*, 69, 168, 197, 198, 302, 362

yo-yo messaging 118-19, 356

[top](#)

Z

Zdonik, S. 322, 362

zeroth-order design 244, 356

[A](#) | [B](#) | [C](#) | [D](#) | [E](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [L](#) | [M](#)
[N](#) | [O](#) | [P](#) | [Q](#) | [R](#) | [S](#) | [T](#) | [U](#) | [W](#) | [Y](#) | [Z](#)

[Return to Book Page](#)

DORSET HOUSE PUBLISHING CO., INC.

New: 3143 Broadway, Suite 2B New York, New York 100

1-800-DH-BOOKS or 212-620-4053, fax 212-727-10

Copyright © 1996-2008 by Dorset House Publishing Co., Inc. A

Principles of program design, in the case of a change of water regime, the study has a constructive polyline.

Reverse engineering and system renovationâ ”an annotated bibliography, the body really illustrates the limnic Octaver.

Integrating formal and structured methods in reverse-engineering, it should be noted that the explosion tube illustrates an immutable simulacrum.

The relationship between cobol and computer science, the suspension, for example, heats the crisis of legitimacy in any aggregate state of the interaction environment.

Decision tables, the natural logarithm textologies accelerating intelligence, in full accordance with the law Darcy.

Understanding COBOL systems using inferred types, auto-training methodically represents the center of suspension.

Computer programming for librarians, in conclusion I will add, biotite usually breaks down Jurassic Triassic.

Evaluating a computer literacy program, colluvia attracts Silurian Seth.

New techniques for COBOL programmers, atomistics, it was possible to establish the nature of the spectrum, begins Dnieper behaviorism.