INDEX

Symbols

#import directive 70, 153
// comment marker 153
@class directive 71, 154, 164
@defs() directive 119, 154, 167
@encode() directive 120–122, 154, 168
@end directive 68, 154, 163, 164
@implementation directive 72, 153, 155, 156, 164
@interface directive 68, 153, 155, 163, 164
@private directive 76–78, 154, 165
@protected directive 76–78, 154, 165
@protocol directive 102, 153, 156, 164
@protocol() directive 104, 154, 156, 168
@public directive 76–78, 154, 165
@selector() directive 81, 154, 168

A

abstract classes 59–60
action messages 84
adopting protocols 102–103, 156
+ alloc method 63, 128
allocating instances 63, 128–131
+ allocFromZone: method 128
anonymous objects 99–100
archiving 149–??
arguments, variable number of 52, 70, 73

B

BOOL data type 152
bycopy type qualifier 113–114, 123, 156, 167

C

C++, using with Objective C 48
categories 94–97
  declaration of 95, 155, 164
  implementation of 95, 155, 164
  uses of 96, 101–102
Class data type 62, 152
@class directive 71, 154, 164
+ class method 62
– class method 62
class methods 62, 69
class objects 55, 61–??, 67
classes 21–24, 55–78
  declaration of 68–72, 155, 163
  implementation of 72–78, 155, 164
_cmd 85, 157
// comment marker 153
conforming to protocols 97, 104, 106–107
– conformsTo: method 104, 106
customizing
  with class objects 63–65

D

@defs() directive 119, 154, 167
designated initializer 135–138
distributed objects 108–114
dynamic binding 32–34, 53–54, 78–81
dynamic loading 34–36, 146–??
dynamic typing 31–32, 50–51

E

encapsulation 25–26
@encode() directive 154, 168, 120–122
@end directive 68, 154, 163, 164
events 39

F

– forward:: method 141–143
forwarding 141–??
– free method 140–141

H

hidden arguments *See* self *and* _cmd

I

id data type 49–51, 60, 116, 152, 157, 166
IMP data type 152
implementation and interface 14–18, 43
@implementation directive 72, 153, 155, 156, 164
#import directive 70, 153
in type qualifier 112–113, 123, 156, 167
information hiding *See* encapsulation
inheritance 27–30, 44, 55–60
– init method 63, 131–138
+ initialize method 66
initializing
  classes 66
  instances 63, 131–140
inout type qualifier 112–113, 123, 156, 167
instance methods 62, 69
instance variables 18, 48–49
  declaration of 69, 155, 165
  inheriting 57–58
  of the receiver 52–53
  outlets 36–38
  referring to 73–78
  scope of 75–78
instances of a class 55
  allocating 63, 128–131
  initializing 63, 131–140
interface and implementation 14–18, 43
@interface directive 68, 153, 155, 163, 164
introspection 50, 61
isa instance variable 50–51, 79
– isKindOf: method 61, 144
– isMemberOf: method 61

L

late binding 34
localization 147–148

M

message expressions 51, 152
message receiver 20, 51, 152
messages 19, 20–21, 51–52, 152
messaging 53–54, 78–81
– methodFor: method 118
methods 18, 48–49
  class methods 62, 69
  declaration of 69–70, 157, 165
  implementation of 73, 157, 165
  inheriting 58–59
  instance methods 62, 69
  of the root class 66–??
  overriding 29, 59
  return and argument types 82, 117

N

naming conventions 158
+ new method 138
nil 50, 52, 153

NXBundle objects 146–??
NXDefaultMallocZone() function 129

O

objc_lookUpClass() function 67
objc_msgSend() function 78
Object class 55, 56–57
objects 18–20, 48–49
oneway type qualifier 110–111, 123, 156, 167
out type qualifier 112–113, 123, 156, 167
outlet instance variables 36–38
overriding methods 29, 59

P

– perform: method 82–83
– perform:with: method 82–83
– perform:with:with: method 82–83
– performv:: method ??–143
polymorphism 26–27, 43, 53
@private directive 76–78, 154, 165
@protected directive 76–78, 154, 165
@protocol directive 102, 153, 156, 164
Protocol objects 103–104
@protocol() directive 104, 154, 156, 168
protocols 97–107
  adopting 102–103, 156
  conforming to 97, 104, 106–107
  declaration of 102–103, 156, 164
  formal 102–107, 156–157
  incorporating other protocols 106–107, 156
  informal 101–102
  type checking 105–106
  uses of 98–101, 109
proxy objects 108–114
@public directive 76–78, 154, 165

R

receiver of a message 20, 51, 152
remote messages 107–114
– respondsTo: method 84, 144
reusability of software 23–24

S

SEL data type 81, 152
sel_getName() function 82
sel_getUid() function 81
@selector() directive 81, 154, 168
selectors 54, 81–82
self 85–89, 90–91, 157, 167
static typing 60–61, 66, 114–118
subclasses 27, 29, 55
super 86–90, 152, 157, 167
superclass 27, 55
surrogate objects 144

T

target-action paradigm 83–84
targets 84
type checking
  class types 116–117
  protocol types 105–106

Z

– zone method 130
zones 129–131