

# Index

- abortive exit 317
- abstract data type 446
- abstract function 515
- abstraction 126
- abstraction 9
- activation record 160
- active block 160
- ad hoc generic 513
- ad hoc polymorphic domain 562
- Ada derived type 457
- Ada generic 532
- additional binding 206
- ADT 446
- alias 52
- allocation process 145
- ambiguity 8
- application 97
- arity 363
- array bounds 400
- array dimension 400
- ASCII 119
- assertion, Prolog 377
- assignment 148
- atom, Prolog 375
- automatic conversion 466
- axioms 365
- backtrack, Prolog 380
- bad restriction 28
- base type of a pointer 398
- base type of a set 408
- base type of an array 400
- BCD 119
- beta redex 100
- beta reducible expression 100
- binary operator 218
- binding a name 180
- binding at source time 530
- binding time 431
- birth of name 197
- birth of object 158
- bitwise operators 27
- block exit time 207
- block structure 195
- block 195
- block 8
- BNF 82
- bound variable 99
- business computer 120
- call-by-address 244
- call-by-constant 247
- call-by-name order 107
- call-by-name 242
- call-by-need 227
- call-by-pointer 251
- call-by-return 247
- call-by-value-and-return 248
- call/cc 322
- call-with-current-continuation 322
- captured reference 225
- checkpoint 322
- class member 500
- clausal logic 366
- closure 236
- closure 260
- coherent assignment 149
- coherent representation 22

- coherent representation 23
- combinatorial explosion 369
- compiler bug 13
- complete name 194
- complete system 96
- complete 368
- comprehension 355
- computation tree 218
- computer representation 20
- conclusion 366
- concrete type 126
- conditional expression 275
- conditional sentence 366
- conditional statement 275
- conformant arrays 527
- conservative conversion 465
- constructor 423
- contents of an object 144
- context free languages 5
- continuation 321
- control diagram 274
- control frame 279
- control frame 289
- control structure 268
- counted loop 293
- curry a function 103
- currying 257
- cut operation, for resolution 370
- cut, Prolog 385
- dangling reference 169
- data hiding 506
- data type 126
- deallocation 158
- death of an object 53
- death of name 197
- death of object 158
- decidable 368
- declaration 179
- deduction 365
- defaults 9
- defining occurrence 195
- demotion cast 462
- demotion 465
- denotational semantics 340
- denotes 339
- dereference 152
- destructive assignment 148
- dictionary 179
- diffuse representation 22
- dispatch a call 517
- dispatch 42
- dispatcher 517
- dispatcher 543
- distinguishable domains 448
- domain checking 444
- domain mapping 450
- domain 436
- dual type 480
- dyadic operator 218
- dynamic binding 180
- dynamic binding 205
- dynamic link 161
- dynamic link 196
- dynamic parent 161
- dynamic scope 198
- dynamic scoping 200
- dynamic storage 159
- EBCDIC 119
- encapsulate 490
- encoding conversion 465
- encoding conversion 466
- enumerated type 396
- environment 179
- error propagation 329
- evaluation, inside-out 224
- evaluation, outside-in 225
- exception handler 331
- exception 327
- expansion of a non-terminal 82
- explicit representation 22
- expressive language 22
- extension 107

- extension 12
- extent of an object 158
- external domain 448
- external object 144
- external scope 194
- first class object 103
- first class objects 52
- first-order predicate calculus 362
- fixed point number 121
- fixed point 121
- fixed point 340
- flexible language 10
- flexible language 28
- formal language 80
- free union 481
- free variable 99
- freelist 164
- front end processor 277
- function domain 57
- function method 517
- function range 58
- functional composition 103
- functional language 148
- functional language 45
- functional 561
- functor 375
- garbage collection 47
- garbage collector 169
- garbage 146
- garbage 165
- general loop 292
- generate a program 85
- generator 356
- generic domain 127
- generic domain 512
- generic function 516
- generic function 523
- generic object 9
- generic package 448
- generic package 532
- generic packages 431
- global scope 194
- goal 378
- good restriction 28
- guarded expression 349
- heap allocation 162
- heap storage 159
- hidden data 492
- higher-order function 103
- higher-order function 254
- Horn clause 372
- hyperresolution 372
- identifier 194
- immortal object 158
- implicit representation 22
- independent domains 448
- indeterminate result 230
- index type 400
- indirectly bound 245
- infinite list 351
- information-losing conversion 465
- inherit 549
- inheritance 42
- inheritance 54
- inherited function 556
- inherited 549
- initialization 146
- in-line code 507
- instance of a generic domain 127
- instantiate a package 431
- instantiate, in C++ 502
- instantiate 168
- instantiate 362
- instantiated type 529
- instantiation 9
- internal domain 448
- internally merged domains 448
- invisible name 198
- IROV 190
- iteration element 298
- labeled scope 64
- lambda calculus formulas 97

- lambda calculus variable 97
- lambda evaluation 226
- lambda expression body 97
- lambda expression parameter 97
- lambda expression 97
- lambda function call syntax 217
- lambda reduction rule 103
- lambda substitution 225
- language extension 12
- lazy evaluation 227
- lexer 59
- lexer generator 60
- lexer 59
- lexical analysis 59
- lexical ancestor 160
- lexical delimiter 59
- lexical parent 160
- lexical scoping 196
- lexical token 59
- lifetime of object 158
- LISP atom 192
- LISP lambda expression 32
- list comprehension 356
- list specification 376
- list 192
- local scope 194
- logic language 47
- logical consequence 368
- logical operators 27
- loop variable 293
- makefile 492
- masked definition 197
- meaning of a name 176
- meaning of a program 90
- meaning of an object 148
- meaning of code 5
- meaning of expression 218
- memory fragmentation 159
- memory management 159
- message 500
- metalanguage 59
- method 42
- method 500
- mode graph 542
- mode 542
- modularity 194
- module 490
- monadic operator 218
- multiple assignment 151
- multiply bound name 180
- name binding 176
- name conflict 225
- name refers to 180
- naming conflict 193
- N*-ary 363
- necessary control structures 344
- nested lifetimes 160
- nonstandardized language 11
- normal exit 317
- normal form 100
- normal form 102
- normal form 99
- normal function call-syntax 217
- normal order 107
- N*-step proof 366
- nuance 10
- object 502
- object-oriented language 193
- object-oriented language 42
- one-in/one-out 271
- opaque derived mode 554
- open list 376
- operands 55
- overloaded name 522
- parallel arrays 25
- parallel evaluation 349
- parallel language 45
- parameter mode 247
- parameterized generic domain 512
- parameterized domain 529
- parameterized module 531

- parameterized type expression 527
- parse tree 218
- parse tree 81
- parse tree 86
- parser generator 80
- parser 80
- partial parameterization 260
- p-code 90
- pointer assignment 154
- pointer variable 146
- polymorphic object 561
- polymorphic type 513
- polymorphic type 561
- portable code 90
- portable program 39
- postfix order 219
- powerful language 28
- predicate 363
- prefix order 219
- prefix syntax 218
- premise 366
- priming read 317
- primitive control 268
- primitive 437
- private symbol 491
- procedural language 42
- process 339
- program object 144
- promises 544
- promotion cast 462
- promotion 465
- proof system 365
- proof 366
- proposition 363
- public symbol 491
- pure functional language 341
- pure value 144
- qualifier list 356
- rank of an array 400
- recurrence equation 339
- redex 100
- reduction 100
- reference constructor 425
- reference conversion 466
- reference 144
- refutation 366
- relatively global scope 194
- representation of an object 20
- represented by 450
- resolution deduction 372
- resolution proof 372
- resolution 366
- resolution 370
- rule, Prolog 376
- run-time dispatcher 543
- run-time stack 160
- satisfy 363
- scientific computer 120
- scope of a name 194
- scope of quantifier 364
- scope 64
- scope-resolution operator 195
- selection function 423
- semantic basis 22
- semantic basis 96
- semantic intent 21
- semantic intent 5
- semantic rules 5
- semantic validity 21
- semiindependent domains 448
- sentence 363
- short circuit evaluation 228
- simple object 128
- size conversion 465
- size conversion 465
- size 128
- slice of an array 403
- spaghetti code 312
- species of a generic domain 127
- specific domain 512
- specific type 126
- specification 338

- stack frame 160
- static binding 180
- static binding 181
- static link 160
- static link 196
- static storage 158
- storage compaction 136
- storage object 144
- stream 94
- strict evaluation 227
- strict evaluation 230
- strict evaluation 350
- strongly typed language 445
- strongly typed language 445
- structure of a program 90
- structured editor 67
- structured programming 271
- subdomain 549
- subroutine call 272
- subroutine return 272
- superdomain 549
- symbol table 179
- syntactic analysis 86
- syntactic category 80
- syntactic sugar 403
- syntax 5
- tail recursion 344
- term 363
- terminated string 406
- theorem 366
- theory 366
- tripcount 297
- tuple 351
- type cast 460
- type cast 460
- type checking 444
- type coercion 460
- type coercion 466
- type compatible 445
- type constructor 453
- type conversion 460
- type conversion 465
- type declaration 126
- type description 126
- type error 445
- type match 445
- type name 126
- type of a type 394
- type predicate 136
- type predicate 431
- type tag 455
- type 137
- type-object 394
- unary operator 218
- unbound name 180
- unbound symbol 225
- undefined name 180
- undefined value 146
- undefined value 295
- unification 370
- union types 419
- unit record equipment 118
- universe of discourse 362
- unlabeled scope 64
- Until test 291
- upward compatibility 39
- use of an identifier 195
- use of parameter 234
- valid representation 21
- valid sentence 366
- value constructor 425
- value of an object 144
- VAR parameter passing 244
- variable declaration 51
- variable 146
- virtual class 554
- virtual function 515
- visibility 552
- visible name 198
- While test 291
- word alignment 412
- zero-based subscripting 400

ZF expression 356