

**Index**

- (negation) operator in constant expressions 127
- operator (subtraction, set difference) in constant expressions 130
- operator in constant expressions 30, 131
- or != operator 330
- Symbols
  - 31, 130, 131, 330
  - ^ operator in constant expressions 31, 131
  - ! (not operator) in constant expressions 30, 126
  - !( boolean\_expression ) operator 329, 332
  - != operator in constant expressions 131
  - !memory operator 329
  - !register operator 329
  - ? command-line option 87
  - @ command-line option 86
  - @@ command-line option 86
  - @a 331
  - @abs function 261
  - @addofs1st function 277
  - @ae 331
  - @align 186
  - @Align procedure option 172
  - @alignstack 186
  - @alignstack procedure option 172, 184, 185
  - @arity function 275
  - @b 331
  - @basetype function 273
  - @basetype function 272
  - @be 331
  - @bound function 278
  - @bound pseudo-variable 417
  - @byte compile-time function 261
  - @byte function 262
  - @c 331
  - @Cdecl procedure option 346
  - @cdecl procedure option 172, 173
  - @ceil function 262
  - @char compile-time function 261
  - @class function 274
  - @cos function 262
  - @cset compile-time function 261
  - @curdir function 277
  - @curlex function 277
  - @curobject function 277
  - @curoffset function 277
  - @date function 262
  - @defined function 275
  - @delete function 265
  - @dim function 275
  - @display 185
  - @display procedure option 172, 184, 185
  - @dword compile-time function 261
  - @e 331
  - @elements function 275
  - @elementsize function 274
  - @enter 186
  - @enter procedure option 174
  - @enumsize function 278
  - @env function 262
  - @EOS function 271
  - @errorprefix pseudo variable 277
  - @eval function 258
  - @exactlynChar function 268
  - @exactlynCset function 267
  - @exactlyniChar function 269
  - @exactlyntomChar function 269
  - @exactlyntomCset function 267
  - @exactlyntomiChar function 269
  - @exceptions function 278
  - @exp function 262
  - @External option (in variable declarations) 33
  - @External procedures 183
  - @extract function 262
  - @firstnChar function 268
  - @firstnCset function 267
  - @firstniChar function 269
  - @floor function 263
  - @FORWARD declarations 184
  - @frame 186
  - @g 331
  - @ge 331

- @index function 265
- @insert function 265
- @int8/@int16/@int32/@int64/@int128  
compile-time functions 261
- @into function 278
- @isalpha function 263
- @isalphanum function 263
- @isclass function 276
- @isconst function 276
- @isdigit function 263
- @IsExternal function 275
- @isfreq function 276
- @islower function 263
- @ismem function 276
- @isreg function 276
- @isreg16 function 276
- @isreg32 function 276
- @isreg8 function 276
- @isspace function 263
- @istype function 276
- @isupper function 263
- @isxdigit function 263
- @l 331
- @lastobject function 277
- @le 331
- @leave 186
- @leave procedure option 174
- @length function 265
- @lex function 274
- @linenumber function 259, 276, 277
- @localoffset function 277, 278
- @locals function 275
- @log function 263
- @log10 function 263
- @lowercase function 265
- @lword compile-time function 261
- @match compile-time function 282,  
296
- @match2 compile-time function 296
- @matchChar compile-time function  
285
- @matchID function 270
- @matchIntConst function 270
- @matchiStr function 269
- @matchNumericConst function 270
- @matchRealConst function 270
- @matchStr function 269
- @matchStrConst function 271
- @matchToiStr function 270
- @matchToStr function 270
- @max function 263
- @min function 263
- @minparmsize function 278
- @na 331
- @nae 331
- @name function 272
- @nb 331
- @nbe 331
- @nc 331
- @ne 331
- @nge 331
- @nl 331
- @nle 331
- @no 331
- @noalignstack 186
- @noalignstack procedure option 172
- @nodisplay 185
- @Nodisplay option 221
- @nodisplay procedure option 172
- @noenter 186
- @noenter procedure option 174, 184,  
185
- @noframe 186
- @noframe procedure option 172
- @noleave 186
- @noleave procedure option 174, 184,  
185
- @nOrLessChar function 268
- @nOrLessCset function 267
- @nOrLessiChar function 269
- @nOrMoreChar function 269
- @nOrMoreCset function 267
- @nOrMoreiChar function 269
- @NOSTORAGE 33
- @ns 331
- @ntomChar function 269
- @ntomCset function 267
- @ntomiChar function 269
- @nz 331
- @o, 331
- @odd function 263
- @offset function 274

- @oneChar compile-time function 285
- @oneChar function 268
- @oneCset function 266
- @oneiChar function 269
- @oneOrMoreChar function 268
- @oneOrMoreCset function 267
- @oneOrMoreiChar function 269
- @oneOrMoreWS function 271
- @optstring function 279
- @parmoffset function 277
- @Pascal procedure option 346
- @pascal procedure option 172, 173
- @pclass function 275
- @peekChar function 268
- @peekCset function 266
- @peekiChar function 269
- @peekWS function 271
- @ptype function 272
- @qword compile-time function 261
- @random function 263
- @randomize function 264
- @read compile-time function 302
- @real32/@real64/@real80 compile-time functions 261
- @REG function 271, 272
- @REG32 function 272
- @REG8 function 271
- @Returns procedure option 172, 346
- @returns procedure option 173
- @rindex function 265
- @s 331
- @section function 280
- @sin function 264
- @size function 274
- @sort function 264
- @sqrt function 264
- @staticname function 274
- @Stdcall procedure option 346
- @stdcall procedure option 172, 173
- @strbrk function 265
- @string
  - operator 280
- @string compile-time function 261
- @strset function 265
- @strspan function 265
- @substr function 265
- @tan function 264
- @text function 280
- @text operator 250
- @time function 264
- @tokenize function 265
- @tostring
  - operator 280
- @tostring operator 257
- @trace function 279
- @trim function 266
- @type function 272
- @typename function 272
- @uns8/@uns16/@uns32/@uns64/
  - @uns128 compile-time functions 261
- @uppercase function 266
- @uptoChar function 268
- @uptoCset function 266
- @uptoiChar function 269
- @uptoiStr function 270
- @uptoStr function 270
- @use procedure option 173
- @use reg32 procedure option 172
- @word compile-time function 261
- @WSorEOS function 271
- @WSthenEOS function 271
- @z 331
- @zeroOrMoreChar function 268
- @zeroOrMoreCset function 267
- @zeroOrMoreiChar function 269
- @zeroOrMoreWS function 271
- @zeroOrOneChar function 268
- @zeroOrOneCset function 266
- @zeroOrOneiChar function 269
- \* (multiplication) operator in constant expressions 128
- \*NIX 54
- / (division) operator in constant expressions 129
- & operator in constant expressions 31, 131
- && operator 329, 332
- #(...)# in macro parameters 250
- #{ ... }# sequence for manually passing parameters 215
- #{...}# parameter quoting mechanism

- 198
- #{...}# sequence (to create thunks) 213
  - #{...}#" code brackets in boolean expressions 333
  - #append 301
  - #asm..#endasm directives 299
  - #closeread compile-time statement 302
  - #closewrite statement 301
  - #else clause 36, 303
  - #elseif clause in #if statement 36, 303
  - #emit directive 299
  - #endif clause 36, 303
  - #endwhile 150
  - #ERROR directive 301
  - #for..#endfor statement 303
  - #if statement 35, 302
  - #Include directive 35, 245, 246
  - #include directive 347
  - #IncludeOnce directive 247
  - #KEYWORD reserved word 252
  - #linker directive 246
  - #match..#endmatch 298
  - #openread compile-time statement 302
  - #openwrite statement 301
  - #PRINT directive 301
  - #regex..#endregex statement 283
  - #return clause 283
  - #system directive 300
  - #TERMINATOR keyword 252
  - #text..#endtext statement 281
  - #while 150
  - #while..#endwhile statement 303
  - #write statement 301
  - + (addition, set union, string concatenation) operator in constant expressions 130
  - = operator in constant expressions 30, 131
  - = or == operator 330
  - == operator in constant expressions 131
  - > operator 330
  - > operator in constant expressions 31, 131
  - >= operator 330
  - >= operator in constant expressions 31, 131
  - >> operator (shift right) in constant expressions 130
  - | operator in constant expressions 31, 131
  - || operator 329, 332
- Numerics
- 80x86 instruction set 36, 397
- A
- aaa instruction 398
  - aad instruction 398
  - aam instruction 398
  - aas instruction 398
  - ABSTRACT keyword 232
  - Abstract methods and abstract base classes 232
  - Accessing fields of a structure 494
  - Accessing the fields of a class 478
  - Accessor methods 469
  - Activation record 112
  - adc instruction 36, 402
  - add instruction 36, 402
  - Addition operator (+) in constant expressions 130
  - Address of a class procedure 239
  - Address of a memory object (calculation) 410
  - Address of a method 239
  - Address of an iterator 239
  - Address-of operator 239
  - align directive 154
  - align directive (in records) 108
  - align procedure option 173
  - Aligning fields within a record 497
  - Allocating objects dynamically 488
  - always exception handling clause 326 and instruction 36, 402
  - AND operator in boolean expressions 332
  - Anonymous records 106
  - Anonymous unions 107
  - ANYEXCEPTION clause in the TRY..ENDTRY statement 325
  - Arithmetic and logical instructions 36, 402

- Arithmetic instructions 38, 408
  - Array constants 118
  - Array data types 28, 105
  - Arrays as structure fields 495
  - Arrays of records 495
  - Art of Assembly Language Programming 5, 7
  - Assembler control compile-time functions 272
  - Automatic code generation in procedures 186
- B**
- b**
    - name command-line option 86
  - Back-end assembler command line parameters 86
  - Back-end assemblers 59
  - Backtracking 294
  - Base classes 484
  - begin..end block 337
  - Big endian data format 42, 419
  - Binary constants 29, 116, 151, 152, 153
  - Binary object file output name ( command-line option) 86
  - Bit scan instructions 42, 419
  - Bit test instructions 42, 419
  - Bitwise type transfer functions 261
  - Block-structured language 218
  - Boolean constants 29, 117
  - Boolean data type 102
  - Boolean expressions for high-level language statements 329
  - bound instruction 41, 416
  - Branch out of range errors 415
  - break and breakif statements 336
  - bsf instruction 419
  - bsr instruction 42, 419
  - bswap instruction 42, 419
  - bt instruction 42, 419
  - btc instruction 42, 419
  - btr instruction 42, 419
  - bts instruction 42, 419
  - Built-in types 151
  - Byte data type 102
- C**
- c command-line option 86
  - Call instruction 192
  - call instruction 412
  - Calling a class procedure 412
  - Calling a procedure 39, 412
  - Calling conventions 348
  - Calling HLA Procedures 192
  - Calling HLA procedures from another language 349
  - Calling methods and class procedures 234
  - Calling procedures written in a different language 349
  - Cascading exceptions 327
  - case clause in switch statement 339
  - Case neutrality 27, 100, 347
  - Case-insensitive character matching (compile-time function) 286
  - cbw instruction 398
  - cd 50
  - CDecl procedure option 348
  - cdq instruction 398
  - Changing the Location of HLA 58
  - char compile-time function 260
  - Character constants 29, 117
  - Character data type 102
  - Character set constants 118
  - Character set data type 102
  - Character set union operator (+) in constant expressions 130
  - Class data types 222
  - Class Methods, Iterators, and Procedures 476
  - Class procedure address 239
  - Class procedures 235
  - Classes 469
  - clc instruction 398
  - cld instruction 398
  - cli instruction 398
  - clts instruction 398
  - cmc instruction 398
  - cmova instruction 40, 415
  - cmovae instruction 40, 415
  - cmovb instruction 40, 415
  - cmovbe instruction 40, 415

- cmovc instruction 40, 415
- cmove instruction 40, 415
- cmovg instruction 40, 415
- cmovge instruction 40, 415
- cmovl instruction 40, 415
- cmovle instruction 40, 415
- cmovna instruction 40, 415
- cmovnae instruction 40, 415
- cmovnb instruction 40, 415
- cmovnbe instruction 40, 415
- cmovnc instruction 40, 415
- cmovne instruction 40, 415
- cmovng instruction 40, 415
- cmovnge instruction 40, 415
- cmovnl instruction 40, 415
- cmovnle instruction 40, 415
- cmovno instruction 40, 415
- cmovnp instruction 40, 415
- cmovns instruction 40, 415
- cmovnz instruction 40, 415
- cmovo instruction 40, 415
- cmovp instruction 40, 415
- cmovpe instruction 40, 415
- cmovpo instruction 40, 415
- cmovs instruction 40, 415
- cmovz instruction 40, 415
- cmp instruction 37, 404
- cmpsb instruction 398
- cmpsd instruction 398
- cmpsw instruction 398
- cmpxchg instruction 41, 417, 418
- Command line parameters 86
- Comments 26, 93
- Comparison operators in constant expressions 131
- Compile only (command-line option) 86
- Compile-time "?" statement
  - ? compile-time statement 146
- Compile-time language 245
- Compiling simple programs with HIDE 4
- Composing instructions 397
- Composite data types 28, 105
- Computing the address of a memory operand 410
- Conditional compilation statements 35, 302
- Conditional jump instructions 40, 415
- Conditional move instructions 40, 415
- Conditional set instructions 40, 415
- Conjunction in boolean expressions 332
- const declaration section 142
- const keyword 142
- Const sections 33
- Constant expressions 30, 124
- Constants
  - array 118
  - binary 29, 116
  - boolean 29, 117
  - character 29, 117
  - character set 118
  - decimal 29, 115
  - \_display\_ 182
  - floating-point 29, 116
  - hexadecimal 29, 115
  - literal 29, 115
  - numeric 29, 115
  - \_parms\_ 182
  - pointer 30, 123
  - record 119
  - regular expression 123
  - string 30, 117
  - structured 30, 118
  - Unicode character 117
  - Unicode string 117
  - union 120
- Constructors 233, 487, 489
- Context free macros 252
- continue and continueif statements 336
- Controlling field offsets within a record 496
- Conversion functions (compile-time language) 260
- cpuid instruction 398
- Create procedure for an object 489
- Creating a New Project in RadASM 41
- Creating a Virtual Method Table 234
- cset compile-time function 260
- Cset data type 102, 152
- Customizing HLA 57
- cwd instruction 398
- cwde instruction 398

## D

daa instruction 398  
 das instruction 398  
 Data Types 27, 102  
 Data types  
   array 28, 105  
   composite 28  
   enumerated 103  
   pointer 111  
   record 28, 106  
   thunks 112  
   union 105  
 dec instruction 38, 408  
 Decimal constants 29, 115  
 Declaration section in an HLA program 135  
 Declarations 32  
   array 105  
   const 142  
   constants 33  
   enum 103  
   external procedure 183  
   forward procedure 184  
   label 135  
   macros 34  
   namespace 167  
   overloaded procedure, iterator, and method 177  
   proc 167  
   procedure 31, 171, 175  
   readonly 165  
   record 106  
   static 33, 160  
   storage 164  
   type 32, 150  
   union 105  
   var 153  
 default clause in switch statement 339  
 Default include file directory 86  
 Deferred macro parameter expansion 258  
 Defining symbols on the HLA command line 86  
 dir 50  
 Disjunction in boolean expressions 332  
 Display (accessing non-local variables)

220

\_display\_array constant 182  
 \_display\_variable 183, 220  
 Displaying messages during compilation 301  
 div instruction 38, 406  
 Div operator in constant expressions 129  
 Divide instructions 38, 406  
 Division operator in constant expressions 129  
 Dot operator 494  
 Dot operator (field name selection) 132  
 Double precision shift instructions 38, 409  
 DUP operator (array constants) 119  
 Dword data type 102  
 dxx command-line option 86  
 dxx=yy command-line option 86  
 Dynamic Object Allocation 488

## E

e  
   name command-line option 86  
 Eager evaluation of macro parameters 258  
 Editing HLA source files within RadASM 55  
 Effective address calculations 410  
 ELF code generation for FreeBSD 87  
 ELF code generation for Linux OS 87  
 ELSE 328  
 ELSEIF statement 328  
 emms instruction 425  
 endconst keyword 142  
 ENDIF 328  
 endlabel keyword 136  
 endswitch clause in switch statement 339  
 endval keyword 147  
 endvar keyword 154  
 enter instruction 41, 417  
 enum 103  
 Enumerated data types 102, 103  
 Errors 301  
 Exception handling in HLA 321

- Exception numbers 323
  - Executable output filename (command-line parameter) 86
  - exit and exitif statements 337, 338
  - EXIT statement 191
  - External compilation units 345
  - External declarations 345
  - External identifiers 27, 100, 101
  - External procedure declarations 183
  - External symbol names (non-HLA identifiers) 346
  - External symbols 27, 100
- F
- f2xm1 instruction 422
  - fabs instruction 422
  - fadd instruction 420
  - faddp instruction 420
  - FASM as back-end assembler (command-line option) 86
  - fasm command-line option 86
  - fbld instruction 420
  - fbstp instruction 420
  - fchs instruction 422
  - fclex instruction 423
  - FCMOVcc instructions 423
  - fcom instruction 422
  - fcomi instruction 423
  - fcomip instruction 423
  - fcomp instruction 422
  - fcompp instruction 422
  - fcos instruction 422
  - fdecstp instruction 423
  - fdiv instruction 421
  - fdivp instruction 421
  - fdivr instruction 421
  - fdivrp instruction 421
  - ffree instruction 423
  - fiadd instruction 421
  - ficom instruction 422
  - ficompp instruction 422
  - fidiv instruction 421
  - fidivr instruction 422
  - Field alignment within a record 497
  - Field Offsets Within a Record 496
  - fild instruction 420
  - fimul instruction 421
  - \_finalize\_string 240
  - Finalizers 240
  - finestp instruction 423
  - finit instruction 422
  - fist instruction 420
  - fistp instruction 420
  - fld instruction 420
  - fldl instruction 422
  - fldcw instruction 423
  - fldl2e instruction 422
  - fldl2t instruction 422
  - fldlg2 instruction 422
  - fldln2 instruction 422
  - fldpi instruction 422
  - fldz instruction 422
  - Floating point constants 116
  - Floating point instructions 42, 420
  - Floating-point constants 29
  - fmul instruction 420
  - fmulp instruction 421
  - fnop instruction 423
  - for..endfor statement 334
  - foreach..endfor statement 342
  - forever..endfor statement 336
  - FORWARD declarations 241
  - Forward procedure declarations 184
  - fpatan instruction 422
  - fprem instruction 422
  - fprem1 instruction 422
  - fptan instruction 422
  - frame procedure option 172, 184, 185
  - freebsd command-line option 87
  - FreeBSD OS ELF code generation 87
  - frndint instruction 422
  - fscale instruction 422
  - fsin instruction 422
  - fsincos instruction 422
  - fsqrt instruction 422
  - fst instruction 420
  - fstcw instruction 423
  - fstp instruction 420
  - fstsw instruction 423
  - fsub instruction 421
  - fsubp instruction 421
  - fsubr instruction 421



- fsubrp instruction 421
  - fst instruction 422
  - fucom instruction 422
  - fucomp instruction 422
  - fucompp instruction 422
  - Function overloading 348
  - fwait instruction 423
  - fxam instruction 422
  - fxch instruction 420
  - fxtract instruction 422
  - fyl2x instruction 422
  - fyl2xp1 instruction 422
- G**
- Gas 1
  - Gas as back-end assembler (command-line option) 86
  - gas command-line option 86
  - gasx command-line option 86
  - Generating a linker response file 86
  - greedy evaluation 295
- H**
- Header files 347
  - Hello World 49
  - Hexadecimal constants 29, 115
  - HIDE 1
    - auto completion 17
    - compiling simple programs 4
    - global settings 11
    - menus 4
    - project file format 18
    - project manager 14
    - project panel 3
    - settings 10
  - High level language statements 321
  - HLA command line options 86
  - hla command-line option 87
  - HLA Compile-Time Language 245
  - HLA customization 57
  - HLA Design Goals 5
  - HLA environment variables 46
  - HLA installation under Windows 45
  - HLA Integrated Development Environment 1
  - HLA internal operation 84
  - HLA language elements 93
  - HLA program format 31
  - HLA source file output (command-line option) 87
  - HLA Standard Library 45, 54
  - HLA type compatibility 104
  - HLA.INI initialization file 77
  - hlabe command-line option 87
  - HLAINC environment variable 48
  - HLALIB environment variable 48
  - hlt instruction 398
  - Human readable source file format (command-line option) 86
  - Hybrid high level boolean expressions 332
  - Hybrid parameter passing in HLA 215, 216
- I**
- i
    - path command-line option 86
  - IDE 24
  - Identifiers 27, 100
  - idiv instruction 38, 406
  - IF statement 328
  - imod instruction 406
  - imul instruction 37, 405
  - in instruction 41, 416
  - IN operator 331
  - in operator 329, 331
  - in operator in constant expressions 131
  - IN reg parameter specification 171
  - inc instruction 38, 408
  - Include file directory 86
  - Include files 35, 246
  - Index operator (selecting an array element) 133
  - Infinite loops 336
  - Inheritance 473, 484
  - Inherited fields in records 107
  - Inheritance 228
  - inherits keyword 107
  - INHERITS keyword (classes) 473
  - INHERITS reserved word 228
  - \_initialize\_string 240
  - Initializers 240

- Input and output instructions 41, 416
  - insb instruction 398
  - insd instruction 399
  - Installation under Windows 45
  - Installing HLA 45
  - Installing HLA under FreeBSD 54
  - Installing HLA under Linux 54
  - Installing HLA under Mac OSX 54
  - Installing RadASM 31
  - Instances (of a class) 471
  - Instruction composition 397
  - insw instruction 399
  - int instruction 41, 416
  - Int128 data type 102
  - Int16 data type 102
  - Int32 data type 102
  - Int64 data type 102
  - Int8 data type 102
  - int8/int16/int32/int64/int128 compile-time functions 260
  - Integrated Development Environments 24
  - Internal name of the HLA main program 86
  - intmul instruction 37, 405
  - into instruction 399
  - invd instruction 399
  - iret instruction 399
  - iretd instruction 399
  - Iterator address 239
  - Iterator declarations (overloaded) 177
  - Iterators 342
- J**
- ja instruction 40, 415
  - jae instruction 40, 415
  - jb instruction 40, 415
  - jbe instruction 40, 415
  - jc instruction 40, 415
  - jcxz instruction 40, 415
  - je instruction 40, 415
  - jecxz instruction 40, 415
  - jg instruction 40, 415
  - jge instruction 40, 415
  - jl instruction 40, 415
  - jle instruction 40, 415
  - jmp instruction 40, 414
  - jna instruction 40, 415
  - jnae instruction 40, 415
  - jnb instruction 40, 415
  - jnbe instruction 40, 415
  - jnc instruction 40, 415
  - jne instruction 40, 415
  - jng instruction 40, 415
  - jnge instruction 40, 415
  - jnl instruction 40, 415
  - jnle instruction 40, 415
  - jno instruction 40, 415
  - jnp instruction 40, 415
  - jns instruction 40, 415
  - jnz instruction 40, 415
  - jo instruction 40, 415
  - jp instruction 40, 415
  - jpe instruction 40, 415
  - jpo instruction 40, 415
  - js instruction 40, 415
  - JT and JF medium level statements 341
  - jz instruction 40, 415
- K**
- Kleene Plus 284
  - Kleene Star 284
- L**
- label declaration section 135
  - label keyword 135
  - lahf instruction 399
  - Lazy (pass by lazy evaluation) parameter option 171
  - Lazy bersus greedy evaluation 295
  - Lazy evaluation parameters 213
  - lea instruction 39, 410
  - leave instruction 399, 417
  - level=h command-line option 87
  - level=l command-line option 87
  - level=m command-line option 87
  - level=v command-line option 87
  - Lex level 219
  - Lexical analysis 309
  - Lexical Scope 218
  - lib
    - path command-line option 86

- license command-line option 86
  - Linker command-line parameters 86
  - Linker response file 86
  - Linking HLA code with other languages 347
  - linux command-line option 87
  - Linux OS ELF code generation 87
  - Literal constants 29, 115
  - Literal record constants 494
  - Little endian data format 42, 419
  - Local symbols in macros 34, 249
  - Local symbols in multi-part macros 255
  - lods instruction 399
  - lods instruction 399
  - lodsw instruction 399
  - Logical AND operator in constant expressions 131
  - Logical instructions 36, 38, 402, 408
  - Logical OR operator in constant expressions 131
  - Logical XOR operation in constant expressions 131
  - Lookahead 292
  - loop instruction 40, 415
  - loope instruction 40, 415
  - loopn instruction 40, 415
  - loopz instruction 40, 415
  - LWord data type 102
  - lxxxxx command-line option 86
- M**
- m command-line option 86
  - Mac OSX/Mach-o code generation 87
  - Mach-o code generation for Mac OS X 87
  - macos command-line option 87
  - Macro invocations 256
  - Macro parameters 256, 257
  - Macros 34, 248
  - Macros as compile-time functions 305
  - main
    - name command-line option 86
  - Make files in RadASM 25
  - Make menu in RadASM 79
  - Mangled names 347
  - Map files 86
- MASM** 1
- MASM as back-end assembler (command-line option) 87
  - masm command-line option 87
  - Memory addressing modes 42, 390
  - Method address 239
  - Method declarations (overloaded) 177
  - Methods 235, 469
  - mkdir 50
  - MMX instructions 42, 423
  - mod instruction 406
  - MOD operator (remainder) in constant expressions 129
  - mov instruction 36, 402
  - movd instruction 424
  - movq instruction 425
  - movsb instruction 399
  - movsd instruction 399
  - movsw instruction 399
  - movsx instruction 39, 411
  - movzx instruction 39, 411
  - mul instruction 37, 404
  - Multidimensional arrays 28, 105
  - Multi-part macros 252
  - Multiplication operator (\*) in constant expressions 128
  - Multiply instructions 37, 404
- N**
- Name (pass by name) parameter option 171
  - Name mangling 347
  - Name parameters 213
  - namespace declaration section 167
  - Naming conventions 347
  - NASM 1
  - NASM as back-end assembler (command-line option) 87
  - nasm command-line option 87
  - neg instruction 38, 408
  - Negated String Matching 287
  - Negation operator in constant expressions 127
  - Nesting record definitions 496
  - New style procedure declarations 175
  - Non-object calls of class procedures 236

- nop instruction 399
  - not in operator 329, 331
  - not instruction 38, 408
  - Not operator (constant expressions) 30, 126
  - Null operand instructions 36
  - Number data types 103
  - Numeric constants 29, 115
  - Numeric data types 103
  - Numeric Set Constants 116
- O
- obj
    - path command-line option 86
  - Object file placement during compilation (command-line option) 86
  - Object Initialization 487
  - Object-oriented programming 222
  - Objects 471
  - operator in constant expressions 133
  - OR operator in boolean expressions 332
  - Ordinal data types 103
  - out instruction 41, 416
  - outsb instruction 399
  - outsd instruction 399
  - outsw instruction 399
  - overload macro 177
  - Overloaded procedure declarations 177
  - overloads keyword 177
  - OVERRIDE keyword 229
  - overrides keyword 108
  - Overriding a method 473
  - Overriding precedence in constant expressions 132
- P
- p
    - path command-line option 86
  - packssdw instruction 424
  - packsswbv 424
  - packuswb instruction 424
  - paddb instruction 423
  - padd instruction 423
  - Padding a record to some number of bytes 498
  - paddsb instruction 423
  - paddsw instruction 423
  - paddusb instruction 423
  - paddusw instruction 423
  - paddw instruction 423
  - pand instruction 424
  - pandn instruction 424
  - Parameters 191
  - Parameters passed on the stack 348
  - Parenthesis in macro parameters 35, 250
  - `_parms_` constant 182
  - Pass by lazy evaluation 213
  - Pass by name parameters 213
  - Pass by reference 203
  - Pass by result parameters 203
  - Pass by value 193
  - Pass by value/result 203
  - Passing by value
    - Byte-Sized Parameters 194
  - Passing byte value
    - Double-word-sized parameters 200
    - Large parameters 202
    - Lword-sized parameters 201
    - Quad-word-sized parameters 200
    - Tbyte-sized parameters 201
    - Word-sized parameters 198
  - Passing parameters in registers 171, 216
  - Path specifications in RadASM 77
  - pavgb instruction 424
  - pavgw instruction 424
  - pcmpeqb instruction 424
  - pcmpeqd instruction 424
  - pcmpeqw instruction 424
  - pcmpgtb instruction 424
  - pcmpgtd instruction 424
  - pcmpgtw instruction 424
  - PE/COFF object code generation 87
  - pextrw instruction 424
  - pinsrw instruction 424
  - PL/360 1
  - PL/M 1
  - pmaddwd instruction 424
  - pmaxsw instruction 424
  - pmaxub instruction 424
  - pminsw instruction 424

- pminub instruction 424
  - pmovmskb instruction 424
  - pmulhuw instruction 424
  - pmulhw instruction 424
  - pmullw instruction 424
  - Pointer constants 30, 123
  - Pointer types 111
  - polymorphism 475
  - pop instruction 39, 411
  - popa instruction 399
  - popad instruction 399
  - popf instruction 399
  - popfd instruction 399
  - por instruction 424
  - Precompiling regular expressions 297
  - Primitive data types 27, 102
  - Private fields in a class 100
  - Private fields in a class 471
  - proc declaration section 167
  - Procedure calls 39, 191, 412
  - Procedure declarations 31, 171
  - procedure declarations 175
  - Procedure declarations (overloaded) 177
  - Procedure options 185
  - Program Structure 31
  - Program structure 134
  - Project organization in RadASM 24
  - Project Panel in HIDE 3
  - Project types in RadASM 78
  - Prototype software 7
  - psadbw instruction 424
  - Pseudo-variables 277
  - pshufw instruction 424
  - pslld instruction 425
  - psllq instruction 425
  - psllw instruction 425
  - psrad instruction 425
  - psraw instruction 425
  - psrld instruction 425
  - psrlq instruction 425
  - psrlw instruction 425
  - psubb instruction 423
  - psubd instruction 424
  - psubsb instruction 424
  - psubsw instruction 424
  - psubusb instruction 424
  - psubusw instruction 424
  - psubw instruction 423
  - punpckhbw instruction 424
  - punpckhdq instruction 424
  - punpckhwd instruction 424
  - punpcklbw instruction 424
  - punpckldq instruction 424
  - punpcklwd instruction 424
  - push and pop instructions 39, 411
  - pusha instruction 399
  - pushad instruction 399
  - pushd instruction 39, 411
  - pushf instruction 399
  - pushfd instruction 399
  - pushw instruction 39, 411
  - \_pVMT\_ 233, 488
  - pxor instruction 424
- Q**
- QWord data type 102
- R**
- r
    - name command-line option 86
  - RadASM execution 31
  - RadASM installation 31
  - RadASM project management 32
  - RADASM.INI file 74
  - RadASM/HLA Integrated Development Environment 24
  - RAISE statement 321
  - raise statement 327
  - Range checking 41, 416
  - rcl instruction 38, 409
  - rcr instruction 409
  - rdmsr instruction 400
  - rdpmc instruction 400
  - rdtsc instruction 400
  - readonly declaration section 165
  - Real (Floating Point) Constants 29
  - Real constants 116
  - Real128 102
  - Real32 data type 102
  - real32/real64/real80 compile-time functions 260

Real64 data type *102*  
 Real80 data type *102*  
 Record constants *119, 494*  
 Record data types *28, 106*  
 Record field alignment *497*  
 Record offsets *111, 496*  
 Records as record fields *495*  
 Recursive file inclusion (prevention)  
     *247*  
 Reference parameters *203*  
 Register parameters *216*  
 Regular expression constants *123*  
 Regular expression macros *281*  
 rep.insb instruction *400*  
 rep.insd instruction *400*  
 rep.insw instruction *400*  
 rep.movsb instruction *400*  
 rep.movsd instruction *400*  
 rep.movsw instruction *400*  
 rep.outsb instruction *400*  
 rep.outsd instruction *400*  
 rep.outsw instruction *400*  
 rep.stosb instruction *400*  
 rep.stosd instruction *400*  
 rep.stosw instruction *400*  
 repe.cmpsb instruction *400*  
 repe.cmpsd instruction *400*  
 repe.cmpsw instruction *400*  
 repe.scasb instruction *401*  
 repe.scasd instruction *401*  
 repe.scasw instruction *401*  
 repeat..until statement *334*  
 repne.cmpsb instruction *401*  
 repne.cmpsd instruction *401*  
 repne.cmpsw instruction *401*  
 repne.scasd instruction *401*  
 repne.scasw instruction *401*  
 Reraising an exception *327*  
 Reserved words *27, 93*  
 Result (pass by result) parameter option  
     *171*  
 Result parameters *203*  
 ret instruction *40, 414*  
 RET with NOFRAME option *189*  
 RETURNS statement *397*  
 returns statement *217*

rol instruction *38, 409*  
 ror instruction *409*  
 Rotate instructions *38, 409*  
 rsm instruction *401*  
 Running HLA *49*  
 Running RadASM *31*

## S

s command-line option *86*  
 sahf instruction *401*  
 sal instruction *38, 409*  
 sar instruction *409*  
 sbb instruction *36, 402*  
 scasb instruction *401*  
 scasd instruction *401*  
 scasw instruction *401*  
 Scoping rules *218*  
 Selecting field names *132*  
 Set intersection operator (\*) in constant  
     expressions *128*  
 Set union operator (+) in constant expres-  
     sions *130*  
 seta instruction *40, 415*  
 setae instruction *40, 415*  
 setb instruction *40, 415*  
 setbe instruction *40, 415*  
 setc instruction *40, 415*  
 sete instruction *40, 415*  
 setg instruction *40, 415*  
 setge instruction *40, 415*  
 setl instruction *40, 415*  
 setle instruction *40, 415*  
 setna instruction *40, 415*  
 setnae instruction *40, 415*  
 setnb instruction *40, 415*  
 setnbe instruction *40, 415*  
 setnc instruction *40, 415*  
 setne instruction *40, 415*  
 setng instruction *40, 415*  
 setnge instruction *40, 415*  
 setnl instruction *40, 415*  
 setnle instruction *40, 415*  
 setno instruction *40, 415*  
 setnp instruction *40, 415*  
 setns instruction *40, 415*  
 setnz instruction *40, 415*

- seto instruction 40, 415
  - setp instruction 40, 415
  - setpe instruction 40, 415
  - setpo instruction 40, 415
  - sets instruction 40, 415
  - Setting auxiliary paths for HLA files 59
  - Setting default procedure options 185
  - setz instruction 40, 415
  - Sevag Krikorian 1
  - Shift and rotate instructions 38, 409
  - Shift left operator ( < ) 130
  - Shift right operator (>>) in constant expressions 130
  - shl instruction 38, 409
  - shld instruction 409
  - shr instruction 409
  - shrd instruction 409
  - Sign and zero extension instructions 39, 411
  - Signed data types 103
  - Signed vs. unsigned comparisons in boolean expressions 330
  - source command-line option 86
  - Source file format output from HLA 86
  - Special symbols and punctuation 26, 93
  - Specifying an external symbols name 346
  - Static class fields 237
  - Static data objects in a class 472
  - static declaration section 160
  - Static member functions 235
  - Static Procedures (in a class) 474
  - Static section 33
  - stc instruction 402
  - std instruction 402
  - Stdcall procedure option 348
  - sti instruction 402
  - storage declaration section 164
  - stosb instruction 402
  - stosd instruction 402
  - stosw instruction 402
  - str.strRec definition 497
  - string compile-time function 260
  - String concatenation operator (+) in constant expressions 130
  - String Constants 30
  - String constants 117
  - String data type 102
  - String representation 497
  - String/Pattern matching compile-time function 266
  - Structure, accessing fields of... 494
  - Structured constants 30, 118
  - Structured goto statement 337
  - Structures as structure fields 495
  - sub instruction 36, 402
  - Subtraction operator (-) in constant expressions 130
  - super keyword 224
  - switch..case..default..endswitch statement 339
  - sym command-line option 87
  - Symbol related compile-time functions 272
  - Symbol table display 90
  - symbol table dump after compile (command-line option) 87
- T
- TASM 1
  - TASM as back-end assembler (command-line option) 87
  - tasm command-line option 87
  - TByte data type 102
  - Temporary working file directory (command-line option) 86
  - test command-line option 87
  - test instruction 36, 402
  - Text data type 103
  - THEN 328
  - THIS 478
  - This (reference to class object) 233
  - thread command-line option 86
  - Thread-safe code generation 86
  - THUNK constants (pass by name/lazy parameters) 213
  - Thunk data type 103
  - Thunks 112
  - try..always..endtry statement 326
  - TRY..EXCEPTION..ENDTRY statement 321

- Type checking *124*
  - Type coercion *44, 125, 394*
  - Type compatibility *104*
  - type declaration section *150*
  - Type promotion *124*
  - Type sections *32*
- U
- UCR Standard Library for 80x86 Programmers *5*
  - ud2 instruction *402*
  - Unicode Character Constants *117*
  - Unicode data type *102*
  - Unicode String Constants *117*
  - Union constants *120*
  - Union data types *105*
  - Units *345*
  - UNPROTECTED clause in the TRY..ENDTRY statement *324*
  - Uns128 data type *102*
  - Uns16 data type *102*
  - Uns32 data type *102*
  - Uns64 data type *102*
  - Uns8 data type *102*
  - Unsigned data types *103*
  - until clause in repeat..until statement *334*
  - Untyped reference parameters *172, 207, 208*
  - User-defined compilation errors *301*
  - User-defined exceptions *323*
- V
- v command-line option *87*
  - Val (pass by value) parameter option *171*
  - val keyword *146*
  - Valres (pass by value/result) parameter option *171*
  - Value parameters *193*
  - Value/Result parameters *203*
  - Var (pass by reference) parameter option *171*
  - VAR (untyped reference parameters) *172*
  - var declaration section *153*
  - var keyword *153*
  - Var type (untyped reference parameters) *208*
  - Variable parameter lists in macros *248*
  - \_vars\_ constant *182*
  - Constants
    - \_vars\_ *182*
  - Verbose compile switch (command-line option) *87*
  - Virtual member functions *235*
  - Virtual method calls *475*
  - Virtual method table *481*
  - Virtual Method Table pointer *233*
  - Virtual Method Tables *482*
  - Virtual method tables *233*
  - Virtual Methods *474*
  - \_VMT\_ *233, 483*
  - VMT *233, 481, 483*
- W
- w command-line option *87*
  - wait instruction *402*
  - wbinvd instruction *402*
  - WChar data type *102*
  - Webster *7*
  - WHILE..ENDWHILE statement *333*
  - win32 command-line option *87*
  - Win32 OS PE/COFF object code generation *87*
  - Windows (GUI) applications *90*
  - Windows API external names *346*
  - Windows installation of HLA *45*
  - Windows Structured Exception Handler *323*
  - Word data type *102*
  - Working files directory (command-line option) *86*
- X
- x
    - name command-line option *86*
  - xadd instruction *42, 418*
  - XCHG instruction *37, 403*
  - xlat instruction *402*
  - xor instruction *36, 402*



Y

Yield *112*

Z

Zero extension instruction *39, 411*

Zero operand instructions *36*

ZString data type *102*