

I

Index

SYMBOLS

(stringizing operator) and macros, 89
(concatenation operator), 101-102
 macros, 89
% (modulus operator), 14
&array_name compared to array_name,
184
***/ symbol (comments), 98**
++ operator (increment operator), 13
++var compared to var++, 13-14
/* symbol (comments), 98
// symbol (comments), 98, 279
32-bit compilers, 356-357
64KB limit for arrays, 150

A

Abort message, 81-83
access
 denying file access, 79
 DOS (Disk Operating System) memory
 locations, 260-262
acos() function, 239
adding
 pointers, 143-144
 values to pointers, 141-142

addresses

array tags, 183-184
base, 178
beyond array endpoints, 177-179
offset, 178
printing, 157-158

algorithms

searching, 32-33
 bsearch() function, 48-50
 complexity, 33-34
sorting, 31-34

Allman brace style, 338

allocating

file handles, 81
memory, 132-158
 heaps, 152-153
 malloc() and calloc(), 149
 sizing, 156
 stacks, 151-152
 Windows, 413

alphabetical characters, 374-375

animated bitmaps, Windows, 403

animated text, writing text to screen, 296

ANSI (American National Standards

Institute), 283-291

 C++ (Windows compiler compatibility),
 395

FREE EBOOKS, NOTES , VIDEOS & PLACEMENT MATERIAL



For All Companies placement
Material

[@placementclasses](#)



For CAT Exam Preparation
Material

[@cat_classes](#)



For GATE Exam Preparation
Material

[@gate_classes](#)



For Engineering Books &
Material

[@cs_ebooks](#)



Codes of Following Programming Languages



C

[@c_examples](#)



C++

[@cpp_examples](#)



Java

[@java_examples0](#)



Python

[@python_examples](#)

- compared to BIOS functions, 264-265
 - ANSI driver, 310-314**
 - ANSI.SYS device driver (escape characters), 382**
 - ANSI/ISO C standard, 110-111**
 - cursor positioning, 295
 - variables
 - declaring, 271
 - environment variables, 256
 - significance for names, 340
 - applicability of command-line parameters, 351-352**
 - applications**
 - compiling, 315-329
 - Ctrl-Break (stopping execution), 300-301
 - data transfer, 363-368
 - directories of executing applications, 368-369
 - fitting to DOS, 324-325
 - interrupting (Windows), 357-360
 - preprocessor, 90-92
 - running, 361-363
 - writing, 315-329
 - argc argument parameter (command-line parameters), 350**
 - arguments, declaring in argument lists, 288-289**
 - argv argument parameter (command-line parameters), 350**
 - arithmetic operations**
 - listing VII.7, 140-141
 - type casts, 26
 - void pointers, 157
 - array_name, compared to &array_name, 184**
 - arrays, 176-187**
 - 64KB limit, 150
 - addressing
 - array tags, 183-184
 - beyond array endpoints, 177-179
 - calloc() function, 149
 - char, 226
 - compared to strings, 186-187
 - constant values, 185
 - lvalues, 11-12
 - navigating with pointers or subscripts, 181-183
 - passing to functions, 167-169
 - pointers, 180
 - sizeof operator, 179-180
 - sizing at runtime, 147-148
 - subscripts, 176-177
 - arrow keys, 304-305**
 - ASCII (American Standard Code for Information Interchange) character set, 189-190**
 - alphabetical characters, 374-375
 - NUL, 155
 - asin() function, 239**
 - assert() function, 212-213**
 - assigning**
 - hexadecimal values to variables, 376
 - octal values to numbers, 377
 - assignment operators, operator precedence, 287**
 - assignment statements, 10-12**
 - associativity, operator precedence, 270-271**
 - atan() function, 239**
 - atan2() function, 239**
 - atexit() function and cleanup operations, 169-170**
 - atoi() function, 126-128**
 - attributes of files, 75-76**
 - AUTOEXEC.BAT file**
 - environment variables, 256
 - printing, 341
-
- B**
- bank switching, memory management, 326**
 - base (addresses), 178**
 - base 2 (binary numbering), 377-379**
 - base 8 (octal numbering), 379**
 - base 16 (hexadecimal numbering), 380-381**
 - BeginPaint() function, 390**
 - big-endian, compared to little-endian, 280-281**
 - binary numbering system, 377-379**
 - compared to text mode, 67
 - searches, 49
 - pointers, 139
 - streams, 67
 - see also* comparison searching
 - BIOS (Basic Input Output System), 255, 262**
 - ANSI functions, 264-265
 - calling functions, 258-260
 - controlling a mouse, 272-273
 - DOS, 262
 - graphics mode, changing, 265-269
 - BIT_POS() macro, 192**
 - BIT_RANGE() macro, 193**
 - BIT_SHIFT() macro, 193**
 - bitmaps, animated (Windows), 403**
 - bits, 189-196**
 - binary numbering, 378
 - masking, 191-194
 - portability of bit fields, 194
 - bitshifting**
 - and multiplication by 2, 194-195
 - operator precedence, 286
 - braces, styles of usage, 338-339**
 - Break key, disabling, 301**
 - break statements**
 - continue statements, 346
 - switch statements, 5
 - breakpoints, setting, 205**
 - bsearch() function, 48-50**
 - buffering output, 293-294**
 - bus errors, 155-156**
 - buttons, 407-408**
 - bytes, 189-196**
 - binary numbering, 378
 - bit masking, 191-194
 - color bytes (writing data to the screen), 298-300
 - high-order bytes, 195
 - low-order bytes, 195
 - bvval_func() function, 167**
-
- C**
- C, 1-14**
 - //for comments, 279
 - ANSI, 110-111, 283-291
 - C++ compiler additions, 277

- comma operator, 6-7
- compared to C++, 277-279
- compiling (`_cplusplus` symbol), 106
- data files, 63-85
- functions, 159-173
- goto statements, 8-10
- ISO, 283-291
- local blocks, 1-3
- `longjmp()` function, 8-10
- loops, 7-8
- lvalues, 10-11
- memory allocation, 132-158
- modulus operator (%), 14
- object-oriented design, 278
- operator precedence, 12-13
- pointers, 132-158
- preprocessor, 87-113
- rvalues, 12
- `setjmp()` function, 8-10
- strings, 116-130
- switch statements, 3-5
- `var++` compared to `++var`, 13-14
- Windows, 385-414
- C Programming Language, The*, 338, 353
- C++**
 - ANSI (Windows compiler compatibility), 395
 - compared to C, 277-279
 - compiler additions in C programs, 277
 - compiling (`_cplusplus` symbol), 106
 - `main()` function, 290
 - object-oriented design, 278
 - Windows SDK, 392
- callback (function pointers)**, 145
- calling**
 - BIOS functions, 258-260
 - DOS functions, 257-258
- calloc() function**, 324
 - arrays, 149
 - compared to `malloc()` function, 149
 - memory leaks, 203
 - zero bits, 149
- CallWindowProc() function**, 409
- camel notation (naming conventions)**, 336
- caption bars**, *see* title bars
- capturing mouse clicks (Windows)**, 402-403
- carets compared to cursors (Windows)**, 401-402
- ceil() function**, 240
- Central Processing Unit (CPU)**, 263
- chaining collisions**, 55
- char arrays**, 226
- character sets (OEM key codes)**, 397-398
- characters**
 - alphabetical characters, 374-375
 - color, printing to screen, 260
 - determining classes of, 232-233
 - field restrictions, 305-307
 - multibyte characters, 240-241
 - numerical characters, 375-376
- chars**, 280
- child windows**, 407-408
- cleanup operations and `atexit()`**, 169-170
- clearing screens with the ANSI driver**, 311
- clicks, capturing (Windows)**, 402-403
- client areas (Windows)**
 - mouse clicks, capturing, 402-403
 - repainting, 395-396
 - sizing, 396-397
- code**
 - commenting out code (preprocessor), 98-99
 - portability, 275-281
 - sort/search examples, 57-62
- collisions, hashing**, 55
- color**
 - screen color (ANSI driver), 312
 - system colors (Windows), 405-406
 - text color (ANSI driver), 312-313
- color bytes, writing data to the screen**, 298-300
- .COM files**, 319
- comma operator**, 6-7
 - operator precedence, 287
- comma-delimited text**, 83-85
- command-line parameters**, 349-352
- commands, File menu, Exit**, 388
- commenting out code (preprocessor)**, 98-99
- comments**, 208
 - `*/` symbol, 98
 - `/*` symbol, 98
 - `//` symbol, 98, 279
 - program efficiency, 333-334
- comp() function**, 37, 49
- Compact memory model**, 317-319
- comparing strings**, 129-130, 228
- comparison searching**, 33
- compile date and time, printing (preprocessor)**, 110
- compile-time checking**, 205-207
- compilers**
 - 32-bit compilers, 356-357
 - warnings, 207
 - Windows compatibility, 394-395
- compiling**
 - C++ (`_cplusplus` symbol), 106
 - programs, 315-329
- complexity**, 33-34
- concatenating strings**, 228
- concatenation operator (##)**, 101-102
 - macros, 89
- conditional expressions**
 - compilation (preprocessor), 91
 - operator precedence, 287
- const modifier**, 21-22, 29
- const pointers**, 18-19
- const_func() function**, 169
- constants**
 - arrays, 185
 - declaring with
 - `const`, 29
 - `#define`, 93-94
 - enum, 94

- enum compared to #define, 95-96
 - system colors (Windows), 406-407
 - type casts, 27
 - volatile variables, 21
 - continue vs. break statements, 346**
 - controls, 407-408**
 - conventions**
 - naming (variables), 332
 - programs, 233
 - converting**
 - date variables to single numbers, 243-247
 - numbers to strings, 124-126
 - strings to numbers, 126-128
 - time variables to single numbers, 248-251
 - cooperative multitasking, 357**
 - copying**
 - string sections, 123-124
 - strings, 228
 - core dumps, 155-156**
 - cos() function, 239**
 - cosh() function, 239**
 - __cplusplus symbol and compiling C++, 106**
 - CPU (Central Processing Unit), 263**
 - CreateDialog() function, 414**
 - CreatePen() function, 390**
 - CreateWindow() function, 388, 407-408**
 - edit class, 410-411
 - listbox class, 411-412
 - creating**
 - buttons (Windows), 407-408
 - .COM files, 319
 - controls (Windows), 407-408
 - delay timers for DOS, 353
 - libraries, 321-322
 - random numbers, 354-356
 - critical error handler (interrupt 24), 81**
 - Ctrl-Alt-Delete (disabling warm boots), 372-374**
 - Ctrl-Break (stopping program execution), 300-301**
 - disabling, 370-372
 - Windows program interrupts, 388
 - ctype.h header file, 232**
 - curses package (cursor positioning), 295**
 - cursors**
 - compared to carets (Windows), 401-402
 - moving (ANSI driver), 313-314
 - positioning, 294-295
 - restoring cursor positions (ANSI driver), 312
 - saving cursor positions (ANSI driver), 311-312
-
- D
- data files, 63-85**
 - Abort, Fail, Retry messages, 81-83
 - attributes, 75-76
 - binary mode, 67
 - comma-delimited text, 83-85
 - date and time lists, 70-72
 - denying access, 79
 - errno variable and nonzero numbers, 63-64
 - file handle allocation, 81
 - file lists in directories, 68-69
 - filename sorting in directories, 73-74
 - locking files, 79-80
 - opening in shared mode, 77-79
 - passing data, 363-368
 - PATH environment variable, 76-77
 - sharing files, 79-80
 - storing data, 16-29
 - streams, 64
 - text mode, 67
 - writing to the screen, 295-296
 - data segments (variable storage), 16**
 - data types and scanf(), 303**
 - __DATE__ preprocessor command, 109-110**
 - date and time, 70-72**
 - compile date and time (preprocessor), 110
 - Windows, 404
 - date variables, 243-253**
 - converting to single numbers, 243-247
 - sorting, 245
 - storing, 243-247, 252
 - dead keys (Windows), 400-401**
 - debugging, 197-213**
 - equality operators, 206
 - excessive execution time, 200-202
 - failed conditions, 212-213
 - hanging programs, 197-203
 - implicitly cast variables, 206-207
 - infinite loops, 199-200
 - input problems, 202-203
 - memory leaks, 203-204
 - methods, 204-211
 - nested for loops, 202
 - symbolic debuggers (enumerated constants), 96
 - tools, 205-207
 - TSR (terminate and stay resident) programs, 211-212
 - uninitialized variables, 206
 - declaring**
 - arguments in argument lists, 288-289
 - array sizes at runtime, 147-148
 - constants with
 - const, 29
 - #decline, 93-94
 - enum, 94
 - functions, 159-162
 - static variables in headers, 28
 - variables, 28, 271
 - in headers, 27
 - local blocks, 2
 - default cases, 4-5**
 - #define directive (preprocessor), 92**
 - constant declarations, 93-94
 - compared to enumerated constants, 95-96
 - macros, 88-89
 - undefining (preprocessor), 111-112
 - true/false, 344
 - defining**
 - headers at compile time (preprocessor), 100
 - NULL as 0, 142

standard library functions, 216-223
 true/false, 344-345
 variables, 27-28

DefWindowProc() function, 402

delay timers
 creating for DOS, 353
 Windows, 387

demo programs, disabling (preprocessor), 97

denying file access, 79

detecting memory leaks, 203-204

device contexts, 386
 GDI (Graphic Device Interface), 389
 HDC (handle), 394

DGROUP: group exceeds 64K message, 323-324

dialog boxes (modal and modeless), 414

DialogBox() function, 414

digital trie searching algorithm, 50-55

directories
 filenames, sorting, 73-74
 listing files, 68-69
 searching for executing programs, 368-369

disabling
 Break key, 301
 Ctrl-Break, 370-372
 demo programs (preprocessor), 97
 warm boots (Ctrl-Alt-Delete), 372-374

disk swapping (memory management), 325

display modes, 265-269
 screen output, 293-294

distribution sorts, 32

DLLs (dynamic link libraries) and Windows, 393

dollars-and-cents values, printing, 307-309

_dos_findfirst() function, 68-69, 369

_dos_findnext() function, 68-69, 369

_dos_getvect() function, 374

DOS (Disk Operating System)
 BIOS, 262
 command-line parameters, 351-352
 delay timers, 353
 fitting applications, 324-325
 functions
 calling, 257-258
 compared to Windows, 392-393
 interrupts 62, command-line parameters, 352
 memory locations, accessing, 260-262
 undocumented functions, 258

DOS Busy Flag, 258

DOS extenders (memory management), 325

double-precision floating point, 239

doubles (numbers), 308

dup() function, 65-66

dynamic linking (Windows), 393

dynamic memory
 allocating, 324
 calloc(), 149
 malloc(), 149
 memory leaks, 203

E

edit class (Windows), 410-411

editing in Windows, 401-402

efficiency of programs
 comments, 333-334
 naming variables, 336-337
 recursion, 343
 white space, 334-336

EMS (expanded memory), 326

enumerated constants
 compared to #define
 (symbolic) constants, 95-96
 declaring constants, 94
 symbolic debuggers, 96
 use with true/false, 344

environment variables, 256-257

environments, free-standing and hosted, 225

equality operators
 compile-time checking, 206
 precedence, 286

errno variable (nonzero numbers), 63-64

error handling
 bus errors, 155-156
 core dumps, 155-156
 debugging, 197-213
 DGROUP: group exceeds 64K message, 323-324
 excessive program execution time, 200-202
 hardware, 81-83
 infinite loops, 199-200
 input problems, 202-203
 memory faults, 155-156
 null pointers, 137
 assignment errors, 155-156
 preventing, 208-211
 printing error locations
 (preprocessor), 104
 source file errors, printing
 (preprocessor), 105

escape characters, 382-383

events, see interrupts

exception handling, 352

exchange sorts, 32

.EXE files, 319-321
 printf() function, 296

exec() function, 361-362

execution
 loops, 7-8
 sequential program execution, 361-362
 simultaneous program execution, 362-363
 stopping with Ctrl-Break, 300-301

Exit command (File menu), 388

exit() function and return statements, 171-173

exiting Windows programs, 388-389

exp() function, 239

expanded memory (EMS), 326

extended display modes, 268

extended memory (XMS), 326

external scope, 165

external sorts, 32, 44-48

F

fabs() function, 240

factorials, calculating, 342-344

- Fail message, 81-83
- false/true, defining, 344-345
- far compared to near, 327-329
- FAR PASCAL declarations (Windows functions), 392
- far pointers, 151
 - compared to near pointers, 150-151
 - writing data to the screen, 298
- fcvt() function, 125-126
- fdopen() function, 65-66
- fflush() function, 294
- Fibonacci numbers, 201
- fields, character restrictions of, 305-307
- __FILE__ preprocessor command, 108
- File menu, Exit command, 388
- filenames, sorting in directories, 73-74
- files
 - Abort, Fail, Retry messages, 81-83
 - attributes, 75-76
 - comma-delimited text, 83-85
 - denying access, 79-85
 - finding, 369-370
 - handles
 - allocating, 81
 - Windows, 387-388
 - listing in directories, 68-69
 - locking, 79-80
 - opening in shared mode, 77-79
 - sharing, 79-80
- filter functions, interrupting programs, 357-360
- fitting applications to DOS, 324-325
- fixed segments (Windows), 412-413
- flags, 190-193
- Flash EPROM chip, 262
- floating-point comparisons, 22-24, 239-240
- floats (numbers), 308
- floor() function, 240
- flushing output buffers, 294
- fmod() function, 240
- for loops, nested, 202
- for statements (comma operators), 6-7
- formatted text, 295
- fprintf() function, 83-85
- free() function, 152-154, 156, 324
 - alternative versions, 236-239
 - memory management, 156-157
- free-standing vs. hosted environments, 225
- freeing pointers twice, 153-154
- freopen() function, 65
- frexp() function, 240
- fscanf() function, 83-85
- functions, 159-173
 - acos(), 239
 - ANSI vs. BIOS, 264-265
 - arguments, declaring in lists, 288-289
 - array passing, 167-169
 - asin(), 239
 - assert(), 212-213
 - atan(), 239
 - atan2(), 239
 - atexit() and cleanup operations, 169-170
 - atoi(), 126-128
 - BeginPaint(), 390
 - BIOS functions, calling, 258-260
 - bsearch(), 48-50
 - byval_func(), 167
 - callback (pointers), 145
 - calloc(), 149, 324
 - memory leaks, 203
 - CallWindowProc(), 409
 - ceil(), 240
 - comp(), 37, 49
 - compared to macros (preprocessor), 98
 - const_func(), 169
 - cos(), 239
 - cosh(), 239
 - CreateDialog(), 414
 - CreatePen(), 390
 - CreateWindow(), 388, 407-408, 410-412
 - debugging, 198-199
 - declaring, 159-162
 - DefWindowProc(), 402
 - DialogBox(), 414
 - DOS functions, calling, 257-258
 - _dos_findfirst(), 68-69, 369
 - _dos_findnext(), 68-69, 369
 - _dos_getvect(), 374
 - dup(), 65-66
 - error prevention, 208-209
 - exec(), 361-362
 - exit(), 171-173
 - exp(), 239
 - fabs(), 240
 - fcvt(), 125-126
 - fdopen(), 65-66
 - fflush(), 294
 - filter functions, interrupting programs, 357-360
 - floor(), 240
 - fmod(), 240
 - fprintf(), 83-85
 - free(), 152-154, 156-157, 324
 - alternative versions, 236-239
 - freopen(), 65
 - frexp(), 240
 - fscanf(), 83-85
 - getch(), 257
 - getche(), 258
 - getenv(), 76-77, 256
 - GetKeyState(), 400
 - GetSysColor(), 405-406
 - GetWindowLong(), 409
 - glob_func(), 160-161
 - GlobalAlloc(), 413-414
 - GlobalFree(), 414
 - GlobalLock(), 414
 - GlobalUnlock(), 414
 - harderr(), 81-83
 - hardresume(), 81
 - hardretn(), 81
 - header files, declaring variables, 271
 - int86(), 257, 263
 - int86x(), 257, 263
 - InvalidateRect(), 395
 - isalnum(), 232
 - isalpha(), 232
 - iscntrl(), 232
 - isdigit(), 232
 - isgraph(), 232
 - islower(), 232
 - isprint(), 232
 - ispunct(), 232

- isspace(), 232
- isupper(), 232
- isxdigit(), 232
- itoa(), 124-126
- jumping out, 233-235
- KbIntProc(), 374
- KeyboardProc(), 360
- keys, 304-305
- KillTimer(), 387
- ldexp(), 240
- libraries, 215-241
 - advantages, 216
 - creating, 321-322
 - defining, 216-223
 - .EXE files, 320-321
 - memory, 229-231
 - strings, 226-229
- localeconv(), 233
- localtime(), 404
- locking(), 78
- log(), 239
- log10(), 239
- longjmp(), 8-10, 233-234
- lseek(), 79-80
- ltoa(), 125
- ltrim(), 119-120
- main(), 45, 289-290
 - prototypes, 271
 - returning values, 272
- malloc(), 16, 149, 324
 - alternative versions, 236-239
 - memory leaks, 203
- mblen(), 241
- mbstowcs(), 241
- mbtowc(), 241
- memchr(), 228, 231
- memcmp(), 231
- memcpy(), 116-117, 231
- memmove, 229, 231
- memrchr(), 228
- memset(), 231
- merge(), 40, 45
- modf(), 240
- naming, 210-211, 337-338
 - camel notation, 336
 - Hungarian notation, 340-341
- NewCommVector(), 367
- open_customer_indexes(), 165-166
- open_customer_table(), 165-166
- parameters, 163-165
- PASCAL-declared functions, 170-171
- PeekMessage(), 357
- pointers, 144-147
- pow(), 240
- print functions, overhead, 297
- print_document(), 104
- print_report(), 163-164
- printf(), 122-123, 128-129, 157, 257, 295-296, 305, 386
- prototyping, 162-163
- putchar(), 295
- _putenv(), 256
- qsort(), 36-37, 73-74, 145-147
- rand(), 354-356
- Rectangle(), 390
- recursion, 342-344
- return statements, 166, 271
 - exit(), 171-173
- rjust(), 120-122
- rtrim(), 117-122
- scanf(), 302-303
- scope, 165
- SelectObject(), 390
- _setargv(), 350
- setjmp(), 8-10, 233-234
- setlocale(), 233
- SetSysColor(), 406
- SetTimer(), 387
- SetupFilters(), 360
- setvbuf(), 294
- SetWindowLong(), 409
- SetWindowText(), 405
- signal(), 235
- sin(), 239
- sinh(), 239
- some_func(), 162-163
- sopen(), 77-79
- sort_files(), 74
- spawn(), 361-362
- split(), 40-41, 45
- sprintf(), 121, 386
- sqrt(), 239
- srand(), 354-356
- stat_func(), 160
- static functions, 165-166
- strcat(), 228
- strchr(), 228
- strcmp(), 37, 129-130, 146-147, 228
- strcoll(), 233
- strcpy(), 116-117, 228
- strcspn(), 228
- stream functions, 68
- strncat(), 226, 228
- strncmp(), 228
- strncpy(), 123-124, 226, 228
- strpbrk(), 228
- strrchr(), 228
- strrev(), 118-120
- strspn(), 228
- strtok(), 228
- strtoul(), 127-128
- system calls, 255
- system(), 362-363
- tan(), 239
- tanh(), 239
- TextOut(), 390, 404
- time(), 404
- timegm(), 251
- timelocal(), 251
- tolower(), 232
- toupper(), 232
- ultoa(), 125
- variable arguments, 223-225
- WaitMessage(), 357
- wcstombs(), 241
- wctomb(), 241
- Windows compared to DOS, 392-393
- wsprintf(), 404

G

- GDI (Graphic Device Interface), 389-390**
- getch() function, 257**
- getche() function, 258**
- getenv() function, 76-77, 256**
- GetKeyState() function and Windows, 400**
- GetSysColor() function and Windows, 405-406**
- GetWindowLong() function, 409**
- glob_func() function, 160-161**
- global heap (Windows), 412**

global scope, 165
global variables, 360
 declaring/defining in headers, 27
 DGROUP: group exceeds 64K message, 323
GlobalAlloc() function and Windows, 413-414
GlobalFree() function and Windows, 414
GlobalLock() function and Windows, 414
GlobalUnlock() function and Windows, 414
goto statements, 8-10
graphics
 changing modes (Basic Input Output System), 265-269
 OEM (Original Equipment Manufacturer) key codes, 397-398

H

.h files (#include statements), 93
HANDLE (Windows), 394
handles (Windows), 387-388
hanging programs, 197-203
harderr() function, 81-83
hardresume() function, 81
hardretn() function, 81
hardware
 error handling, 81-83
 interrupts, 263
hashing (searching algorithms), 33, 55-57
HDC (Windows), 394
header files
 ctype.h, 232
 declaring (variables), 271
 defining at compile time (preprocessor), 100
 math.h, 239
 redundancy (preprocessor), 92
 setjmp.h, 234
 signal.h, 235
 standard library functions, 216-223
 static variable declarations, 28
 stdarg.h, 223
 stddef.h, 240

stdlib.h, 236
 string.h, 226
 variable declarations/definitions, 27
 windows.h, 344
heaps (memory), 152-153
 far heaps, 327-329
 near heaps, 327-329
 recursion, 152
 strings, 152
 variable storage, 16
 Windows, 412-413
hexadecimal numbering system, 380-381
 assigning values to variables, 376
high-order bytes, 195
hosted vs. free-standing environments, 225
Huge memory model, 317-319
Hungarian notation (naming conventions), 333, 340-341
HWND (Windows), 394

I

if statements
 multiple if statements and switch statements, 3-4
 pointers, 143
#ifdef directive (preprocessor), 112
 portability, 276
#ifndef directive (preprocessor), 92, 112
implicitly cast variables (compile-time checking), 206-207
#include <file> compared to #include "file", 99
#include statements (.h files), 93
include files, nesting (preprocessor), 100-101
increment operator (++) operator, 13
incremented variables
 infinite loops, 199
 passing to macros, 88-89
indirection (pointers), 133
 null pointers, 135-138
InDos Flag, 258

infinite loops, 199-200
 null loops, 345-346
initializing variables, 16-17
input and scanf(), 302-303
insertion sorts, 31
int86() function, 257, 263
int86x() function, 257, 263
integers
 converting strings to integers, 126-128
 converting to strings, 124-126
 mathematical operations, 239-240
integral types (mathematical operations), 24
internal sorts, 32
interrupt 24 (critical error handler), 81
interrupt 62 (command-line parameters), 352
interrupting Windows programs, 357-360, 388-389
interrupts, 263-264
ints, 280
InvalidateRect() function, 395
isalnum() function, 232
isalpha() function, 232
iscntrl() function, 232
isdigit() function, 232
isgraph() function, 232
islower() function, 232
ISO (International Standards Organization), 283-291
isprint() function, 232
ispunct() function, 232
isspace() function, 232
isupper() function, 232
isxdigit() function, 232
iterative processing, 341-342
itoa() function, 124-126

J-K

jmp_buf variable, 234
jumping out (functions), 233-235
KbIntProc() function, 374
Kernighan and Ritchie brace style, 338
Kernighan, Brian W., 353
keyboard (dead keys), 400-401

KeyboardProc () function, 360
keystroke processing with
scanf (), 303
KillTimer () function and
Windows, 387

L

Large memory model, 317-319
ldexp () function, 240
leading spaces (strings),
118-120
leaks in memory, detecting,
203-204
letter characters, 374-375
levels of pointers, 134-135
libraries, 215-241
 creating, 321-322
 cursor positioning, 294
 functions
 advantages, 216
 defining, 216-223
 .EXE files, 320-321
 memory, 229-231
 multiple functions in one
 source file, 321
 reusable functions,
 321-322
 source files, 320
 strings, 226-229
 printing, 297
#line directive (preprocessor),
107-108
_ _LINE_ _ preprocessor
command, 108
linear searches, 50
linked lists
 recursion, 136
 searching, 57
 sorting, 57
lint (debugging tool), 205
listbox class (Windows),
411-412
listing
 date and time of files, 70-72
 files in directories, 68-69
listings
 III.1. qsort(), 36-37
 III.2a. Quick sort, 38-40
 III.2b. Merge sort, 41-42
 III.2c. Radix sort, 43-44
 III.3. External sorting

algorithm, 45-48
 III.4a. bsearch(), 49
 III.4b. Binary searches, 49-50
 III.4c. Linear searching, 50
 III.5. Digital trie searching,
 51-55
 III.6. Hash algorithm, 56
 III.9. Building programs, 59
 III.9a. driver1.c driver , 60
 III.9b. driver2.c driver, 60
 III.9c. driver3.c driver, 61-62
 III.9d. list.h header file, 61-62
 III.9e. list.c source file, 61-62
 III.9f. hash.h header file, 62
 III.9g. hash.c source file, 62
 VII.1. Indirection, 133
 VII.2. Circular list with
 infinite indirection, 134-135
 VII.7. Pointer arithmetic,
 140-141
 VII.15. Arrays with runtime
 size, 147-148
 X.2. Macros for handling flags,
 192
 X.4. Bitshifting and
 multiplication by 2, 195
 XII.3. printf-like function,
 223-225
 XII.5a. string-n functions, 227
 XII.5b. strtok, 228-229
 XII.6. Moving data, 229
 XII.9. setjmp() and
 longjmp(), 234-235
 XII.12. Pool allocator,
 237-238
literals (string literals), 186-187
little-endian compared to
big-endian, 280-281
local blocks, 1-3
local scope, 165
localeconv () function, 233
locales, 233
localtime () function and
Windows, 404
locating
 directories of executing
 programs, 368-369
 files, 369-370
locking files, 79-80
locking () function, 78
log () function, 239

log10 () function, 239
logic checking (default cases), 5
logical operators (operator
precedence), 286
longjmp () function, 8-10,
233-234
longs, 280
loops
 break vs. continue statements,
 346
 error handling, 209-210
 executing successfully, 7-8
 infinite loops
 debugging, 199-200
 vs. null loops, 345-346
 iterative processing, 341-342
 nested for loops, debugging,
 202
 null loops, 345
low-order bytes, 195
lseek () function, 79-80
ltoa () function, 125
ltrim () function, 119-120
lvalues, 10-12

M

macros
 BIT_POS(), 192
 BIT_RANGE(), 193
 BIT_SHIFT(), 193
 compared to functions
 (preprocessor), 98
 concatenation operator (##),
 89, 101-102
 #define statement, 88-89
 flag handling (listing X.2), 192
 incremented variables, passing,
 88-89
 NDEBUG, 213
 predefined macros
 (preprocessor), 103
 preprocessor, 88-89
 SET_FLAG(), 193
 SET_MFLAG(), 193
 stringizing operator (#), 89
 type-insensitive macros
 (preprocessor), 102-103
 undefining (preprocessor),
 111-112
main () function, 45
 C++, 290

- prototypes, 271, 289-290
 - returning values with, 272, 290
 - make utilities, 322-323**
 - makefiles (sort/search sample code), 58**
 - malloc () function, 324**
 - alternative versions, 236-239
 - compared to calloc () function, 149
 - memory leaks, 203
 - variable storage, 16
 - masking bits, 191-194**
 - math.h header file, 239**
 - mathematical operations**
 - floating-point types, 24, 239-240
 - integers, 239-240
 - integral types, 24
 - operator precedence, 286
 - pointer types, 24
 - type casts, 26
 - variables, 24-25
 - void pointers, 157
 - mblen () function, 241**
 - mbstowcs () function, 241**
 - mbtowc () function, 241**
 - Medium memory model, 317-319**
 - memchr () function, 228, 231**
 - memcmp () function, 231**
 - memcpy () function, 231**
 - compared to strcpy (), 116-117
 - memmove () function, 229, 231**
 - memory**
 - allocating, 132-158
 - calloc (), 149
 - malloc (), 149
 - memory leaks, 203
 - pool allocators, 237, 239
 - recursion, 343
 - Windows, 413
 - data transfer, 363-368
 - DOS locations, 260-262
 - dynamic memory allocation, 324
 - faults, 155-156
 - fixed segments (Windows), 412-413
 - heaps, 152-153
 - leaks, 203-204
 - managing, 324
 - bank switching, 326
 - disk swapping, 325
 - DOS extenders, 325
 - EMS (expanded memory), 326
 - free () function, 156-157
 - near and far, 327-329
 - overlay managers, 324-325
 - XMS (extended memory), 326
 - movable segments (Windows), 412-413
 - organizing (Windows), 412-413
 - page thrashing, 17-18
 - raw memory (void pointers), 138-139
 - sizing allocated memory, 156
 - stacks, 151-152
 - variable storage, 16
 - memory image files, see .COM files**
 - memory mapped hardware (volatile modifier), 20-21**
 - memory models, 317-319**
 - memrchr () function, 228**
 - memset () function, 231**
 - merge sorts, 32, 40-42**
 - merge () function, 40, 45**
 - MMU (Memory Management Unit), 17**
 - modal and modeless dialog boxes, 414**
 - modf () function, 240**
 - modular programming, 316**
 - modulus operator (%), 14**
 - monetary values, printing, 307-309**
 - monitor programs, writing text to the screen, 297**
 - Motherboard BIOS, 262**
 - mouse**
 - click capturing (Windows), 402-403
 - controlling (BIOS), 272-273
 - interrupt services, 272
 - Windows, 401-402
 - movable segments (Windows), 412-413**
 - moving cursor positions (ANSI driver), 313-314**
 - multibyte characters, 240-241**
 - multiple if statements (switch statements), 3-4**
 - multiple library functions in one source file, 321**
 - multiplication by 2 and bitshifting, 194-195**
-
- ## N
- namings**
 - functions, 210-211, 337-338
 - Hungarian notation, 340-341
 - variables, 210-211, 236, 339-340
 - ANSI/ISO C standard, 340
 - camel notation, 336
 - Hungarian notation, 333
 - indicating data type, 332-333
 - program efficiency, 336-337
 - underscores, 332
 - natural (sorting algorithms), 32**
 - navigating arrays with pointers or subscripts, 181-183**
 - NDEBUG macro, 213**
 - near compared to far, 327-329**
 - near pointers compared to far pointers, 150-151**
 - nested for loops, debugging, 202**
 - nesting include files (preprocessor), 100-101**
 - network byte order, 281**
 - NewCommVector () function, 367**
 - nibbles (binary numbering), 378**
 - non-English characters, see multibyte characters**
 - nonzero numbers (errno variable), 63-64**
 - null loops vs. infinite loops, 345-346**
 - null pointers, 135-138**
 - assignment errors, 155-156
 - null terminators (strings), 306**

NULLs

compared to NULs, 155
defining as 0, 142

numbers

binary numbering, 377-379
characters, 375-376
converting strings to numbers, 126-128
converting to strings, 124-126
doubles, 308
floats, 308
hexadecimal numbering, 380-381
octal numbering, 377, 379
scientific notation, 310
variables, maximum values of, 23-24
zero-padding, 307

O

object-oriented design (C and C++), 278

octal values, 379

assigning to numbers, 377

OEM (Original Equipment Manufacturer) key codes, 397-398

offset (addresses), 178

open addressing (collisions), 55

open_customer_indexes() function, 165-166

open_customer_table() function, 165-166

operator precedence, 12-13, 269-271, 284-287

assignment operators, 287
associativity, 270-271
bitwise shifting, 286
comma operators, 287
conditional expressions, 287
equality comparisons, 286
logical operators, 286
mathematical expressions, 286
parentheses, 286
postfix expressions, 285
prefix expressions, 285
relational comparisons, 286
 $x=y=z$, 287

operator promotion, 25-26

output, 293-294

overflow errors

(string conversion), 127-128
overhead (print functions), 297
overlay managers (memory management), 324-325
overriding defined macros (preprocessor), 111-112

P

packages, printing, 297
padding strings to fixed lengths, 122-123

page faults, 17

page thrashing, 17-18

paging out, 17

parameters (functions), 163-165

parentheses (operator precedence), 286

pascal calling convention (Windows functions), 392

PASCAL-declared functions, 170-171

passing

arrays to functions, 167-169
data, 363-368
incremented variables to macros, 88-89

PATH environment variable, viewing, 76-77

PeekMessage() function, 357

PMM (Process Memory Map), 17

pointers, 132-158

adding, 143-144
adding values to, 141-142
arithmetic (listing VII.7), 140-141
arrays, 180
binary searches, 139
callback (function pointers), 145
far pointers, 150-151, 298
freeing pointers twice, 153-154
functions, 144-147
if statements, 143
indirection, 133
levels of pointers, 134-135
mathematical operations, 24
memory models, 132
navigating arrays, 181-183

near pointers, 150-151

null pointers, 135-138

NULLs compared to NULs, 155

portability, 133

subtracting, 139-141

to const, 18-19

void pointers, 138-139

writing data to the screen, 298

pool allocators, 237, 239

portability, 275-281

// for comments in C, 279

big-endian compared to little-endian, 280-281

bit fields, 194

C++ compiler additions in C programs, 277

chars, 280

#ifdefs, 276

ints, 280

longs, 280

pointers, 133

shorts, 280

time standards, 251

postfix operations, 13-14

operator precedence, 285

pow() function, 240

#pragma directive (preprocessor), 106-107

predefined macros (preprocessor), 103

prefix operations, 13-14
operator precedence, 285

preprocessor, 87-113

ANSI C standard, 110-111

commenting out code, 98-99

compile date and time, printing, 110

concatenation operator (##), 101-102

conditional compilation, 91

constant declarations

#decline, 93-94

enum, 94

__DATE__ preprocessor command, 109-110

#define directive, 92

use with true/false, 344

demo programs, disabling, 97

error locations, printing, 104

__FILE__ preprocessor command, 108

- header files
 - defining at compile time, 100
 - redundancy, 92
 - `#ifdef` directive, 112
 - `#ifndef` directive, 92, 112
 - `#include <file>` compared to `#include "file"`, 99
 - `#include` statements (.h files), 93
 - include files, nesting, 100-101
 - `#line` directive, 107-108
 - `__LINE__` preprocessor command, 108
 - macros, 88-89
 - compared to functions, 98
 - `#pragma` directive, 106-107
 - predefined macros, 103
 - programs, 90-92
 - source files, printing
 - errors, 105
 - line numbers, 109
 - names, 108
 - symbolic constants, 91
 - symbols, checking for
 - definition, 112
 - `__TIME__` preprocessor command, 109-110
 - type-insensitive macros, 102-103
 - preventing errors, 208-211**
 - print functions (overhead), 297**
 - `print_document()` function, 104
 - `print_report()` function, 163-164
 - `printf()` function, 122-123, 128-129, 157, 257
 - executables, 296
 - Windows, 386
 - writing data to the screen, 295-296, 305
 - printing**
 - addresses, 157-158
 - AUTOEXEC.BAT file, 341
 - color characters to screen, 260
 - compile date and time (preprocessor), 110
 - dollars-and-cents values, 307-309
 - error locations (preprocessor), 104
 - file attributes, 75-76
 - libraries, 297
 - packages, 297
 - PATH environment variable, 76-77
 - redirection (stdout), 66-67
 - scientific notation, 310
 - source files
 - errors, 105
 - line numbers, 109
 - names, 108
 - string sections, 128-129
 - Process Memory Map (PMM), 17**
 - Program Segment Prefixes (PSPs) and command-line parameters, 349-351**
 - programs**
 - compiling, 315-329
 - conventions, 233
 - Ctrl-Break (stopping execution), 300-301
 - data transfer, 363-368
 - debugging, 197-213
 - failed conditions, 212-213
 - methods, 204-211
 - tools, 205-207
 - TSR (terminate-and-stay-resident) programs, 211-212
 - directories of executing programs, 368-369
 - efficiency, 333-336
 - fitting to DOS, 324-325
 - hanging programs, 197-203
 - debugging, 198-199
 - excessive execution time, 200-202
 - infinite loops, 199-200
 - waiting for input, 202-203
 - interrupting (Windows), 357-360
 - modular programming, 316
 - preprocessor, 90-92
 - sequential execution, 361-362
 - simultaneous execution, 362-363
 - speed of execution
 - searching algorithms, 33-35, 50-55
 - sorting algorithms, 33-35, 37-44
 - writing, 315-329
 - prototyping**
 - functions, 162-163
 - `main()` function, 271, 289-290
 - pseudo-random number generators, 354-356**
 - PSPs (Program Segment Prefixes) and command-line parameters, 349-351**
 - `putchar()` function, 271
 - `_putenv()` function, 256
-
- Q-R**
- `qsort()` function, 36-37, 73-74, 145-147
 - quick sort, 38-40**
 - radix searching, 33**
 - radix sort, 32, 43-44**
 - `rand()` function, 354-356
 - random number generation, 354-356**
 - ranges and date variables, 244**
 - raw memory (void pointers), 138-139**
 - `Rectangle()` function, 390
 - recursion, 342-344**
 - heaps, 152
 - linked lists, 136
 - null pointers, 136-138
 - redirection**
 - printing stdout, 66-67
 - standard streams, 65-66
 - refreshing windows, 395-396**
 - register modifier, 19-20**
 - relational comparisons (operator precedence), 286**
 - repainting client areas (Windows), 395-396**
 - residency checks (data transfer), 368**
 - resource editors (buttons and controls), 407**
 - restoring**

- cursor positions (ANSI driver), 312
 - redirected standard streams, 65-66
 - retrieving environment variables, 256-257**
 - Retry message, 81-83**
 - return statements**
 - exit() function, 171-173
 - void functions, 166
 - return() function, 271**
 - returning values with main(), 290**
 - reusable functions (libraries), 321-322**
 - RGB values (Windows system colors), 406**
 - right-justification of strings, 120-122**
 - Ritchie, Dennis M., 353**
 - rjust() function, 120-122**
 - rtrim() function, 117-122**
 - running programs**
 - sequentially, 361-362
 - simultaneously, 362-363
 - rvalues, 12**
- S**
-
- saving**
 - cursor positions (ANSI driver), 311-312
 - windows, 395-396
 - scan statements (strings), 306**
 - scanf() function, 302-303**
 - scientific notation, printing, 310**
 - scope (functions), 165**
 - screens**
 - clearing with the ANSI driver, 311
 - color (ANSI driver), 312
 - output, 293-294
 - SDK (Software Development Kit)**
 - C++, 392
 - Windows, 391-392
 - searching algorithms, 32-33**
 - binary searching, 49
 - bsearch() function, 48-50
 - comparison searching, 33
 - complexity, 33-34
 - digital trie, 50-55
 - hashing, 33, 55-57
 - linear searching, 50
 - linked lists, 57
 - radix searching, 33
 - sample codes, 57-62
 - sequential searching, 33
 - speed of execution, 33-35, 50-55
 - secondary data storage (sorting algorithms), 44-48**
 - segmented architecture, 179**
 - selection sorts, 32**
 - SelectObject() function, 390**
 - sentinel values (null pointers), 137-138**
 - sequential searching, 33**
 - _setargv() function, 350**
 - SET_FLAG() macro, 193**
 - SET_MFLAG() macro, 193**
 - setjmp() function, 8-10, 233-234**
 - setjmp.h header file, 234**
 - setlocale() function, 233**
 - SetSysColor() function and Windows, 406**
 - SetTimer() function and Windows, 387**
 - setting**
 - breakpoints, 205
 - watches, 205
 - SetupFilters() function, 360**
 - setvbuf() function, 294**
 - SetWindowLong() function, 409**
 - SetWindowText() function, 405**
 - shared files, 79-80**
 - shared memory (volatile modifier), 20-21**
 - shared mode (opening files), 77-79**
 - shorts, 280**
 - signal handlers, 235**
 - signal() function, 235**
 - signal.h header file, 235**
 - signals, 235-236**
 - disabling Ctrl-Break, 371-372
 - sin() function, 239**
 - sinh() function, 239**
 - sizeof operator (arrays), 179-180**
 - sizing**
 - allocated memory, 156
 - arrays
 - at runtime, 147-148
 - constant values, 185
 - client areas (Windows), 396-397
 - Small memory model, 317-319**
 - Software Development Kit (SDK)**
 - C++, 392
 - Windows, 391-392
 - software interrupts, 263**
 - some_func() function, 162-163**
 - sopen() function, 77-79**
 - sort_files() function, 74**
 - sorting**
 - date variables, 245
 - filenames in directories, 73-74
 - linked lists, 57
 - sorting algorithms, 31-32**
 - complexity, 33-34
 - distribution sorts, 32
 - exchange sorts, 32
 - external sorts, 32, 44-48
 - insertion sorts, 31
 - internal sorts, 32
 - merge sorts, 32, 40-42
 - natural, 32
 - qsort() function, 36-37
 - quick sorts, 38-40
 - radix sorts, 43-44
 - sample codes, 57-62
 - selection sorts, 32
 - speed of execution, 33-35, 37-44
 - stable, 32
 - source code**
 - comments, 334
 - modular programming, 316
 - source files**
 - errors, printing, 105
 - library functions, 320
 - line numbers, printing, 109
 - multiple library functions, 321
 - names, printing, 108
 - writing programs, 316
 - spaces in strings, 129**
 - spawn() function, 361-362**

speed of execution

- searching algorithms, 33-35, 50-55
- sorting algorithms, 33-35, 37-44
- split() function**, 40-41, 45
- sprintf() function**, 121
 - Windows, 386
- sqrt() function**, 239
- rand() function**, 354-356
- stable (sorting algorithms)**, 32
- stacks (memory)**, 151-152
 - variable storage, 16
- standard display modes**, 267
- standard library functions**, 215-241
 - advantages, 216
 - defining (header files), 216-223
 - memory, 229-231
 - strings, 226-229
- standard predefined macros (preprocessor)**, 103
- standard streams**, 65-66
- standards**
 - ANSI, 283-291
 - ISO, 283-291
 - time, 251
- stat_func() function**, 160
- statements, break and continue**, 346
- static child windows**, 408
- static functions**, 165-166
- static scope**, 165
- static variables**, 360
 - declaring in headers, 28
- stdarg.h header file**, 223
- stdaux stream**, 64
- __STDC__ (ANSI C standard)**, 110-111
- stddef.h header file**, 240
- stderr stream**, 64
- stdin stream**, 64
- stdlib.h header file**, 236
- stdout stream**, 64
 - print redirection, 66-67
- stdprn stream**, 64
- storing**
 - data, 16-29
 - date variables, 243-247, 252
 - flags, 190-191

- time variables, 248-253
- variables, 16
 - in local blocks, 3
- strcat() function**, 228
- strchr() function**, 228
- strcmp() function**, 37, 129-130, 146-147, 228
- strcoll() function**, 233
- strcpy() function**, 228
 - compared to memcpy(), 116-117
- strcspn() function**, 228
- streams**, 64, 68
 - binary streams, 67
 - redirecting, 65
 - restoring redirected standard streams, 65-66
 - scanf() function, 302
 - stdaux, 64
 - stderr, 64
 - stdin, 64
 - stdout, 64
 - print redirection, 66-67
 - stdprn, 64
 - text streams, 67
- string literals**, 186-187
- string.h header file**, 226
- stringizing operator (#) and macros**, 89
- strings**, 116-130
 - compared to arrays, 186-187
 - comparing, 129-130, 228
 - concatenating, 228
 - converting
 - numbers to strings, 124-126
 - to numbers, 126-128
 - copying, 228
 - sections of, 123-124
 - heaps, 152
 - leading spaces, 118-120
 - manipulating, 226-229
 - multibyte characters, 241
 - null terminators, 306
 - overflow errors (converting strings to numbers), 127-128
 - padding strings to fixed lengths, 122-123
 - printing string sections, 128-129
 - right-justification, 120-122

- scan statements, 306
- spaces, 129
 - trailing spaces, 117-118, 129
- strncat() function**, 226, 228
- strncpy() function**, 228
- strncpy() function**, 123-124, 226, 228
- strpbrk() function**, 228
- strchr() function**, 228
- strrev() function**, 118-120
- strspn() function**, 228
- strtok() function**, 228
- strtoul() function**, 127-128
- structured exception handling**, 352
- subclassing windows**, 409
- subscripts (arrays)**, 176-177
 - navigating arrays, 181-183
- subtracting pointers**, 139-141
- switch statements**, 3-5
- symbolic constants**
 - compared to enumerated constants, 95-96
 - preprocessor, 91
- symbolic debuggers (enumerated constants)**, 96
- symbols, checking for definition (preprocessor)**, 112
- system calls**, 255
- system colors (Windows)**, 405-407
- system() function**, 362-363

T

- tags, array (addressing)**, 183-184
- tan() function**, 239
- tanh() function**, 239
- terminate-and-stay-resident programs**, *see* **TSR programs**
- terminating Windows programs**, 388-389
- termination handling (try-finally statement)**, 352
- text**
 - animated text, 296
 - color
 - ANSI driver, 312-313
 - writing data to the screen, 298-300
 - comma-delimited text, 83-85

formatted text, 295
 monitor programs, 297
 writing to the screen, 296-300

text editors (writing text to the screen), 296

text mode compared to binary mode, 67

text streams, 67

TextOut() function, 390, 404

_ _TIME_ _ preprocessor command, 109-110

time and date, 70-72
 compile time and date (preprocessor), 110
 Windows, 404

time variables, 243-253

time() function and Windows, 404

timegm() function, 251

timelocal() function, 251

timer events (animated bitmaps), 403

timers (Windows), 387

Tiny memory model, 317-319

title bars, updating (Windows), 405

tolower() function, 232

tools for debugging, 205-207

toupper() function, 232

trailing spaces (strings), 117-118, 129

transferring data, 363-368

transistors (base 2-binary), 377-379

true/false, defining, 344-345

try-except statement (exception handling), 352

try-finally statement (termination handling), 352

TSR (terminate-and-stay-resident) programs
 data transfer, 364-368
 debugging, 211-212

type casts, 26-27

type-insensitive macros (preprocessor), 102-103

U

ultoa() function, 125

undefining macros (preprocessor), 111-112

underscores
 camel notation, 336
 variable names, 236, 332

uninitialized variables (compile-time checking), 206

unsigned variables (infinite loops), 200

updating title bars (Windows), 405

V

values, returning with main(), 272, 290

var++ compared to ++var, 13-14

variables, 16-29
 arguments (functions), 223-225
 const modifier, 21-22
 date, 243-253
 declaring, 28, 271
 in headers, 27
 defining, 28
 in headers, 27
 environment variables, retrieving, 256-257
 global variables, 360
 hexadecimal value assignments, 376
 implicitly cast variables (compile-time checking), 206-207
 initializing, 16-17
 jmp_buf, 234
 local blocks, 2
 mathematical operations, 24-25
 naming, 210-211, 236, 339-340
 ANSI/ISO C standard, 340
 camel notation, 336
 Hungarian notation, 340-341
 indicating data type, 332-333
 program efficiency, 336-337
 underscores, 332
 numeric variables, maximum values of, 23-24
 register modifier, 19-20
 static variables, 360
 storing, 16
 in local blocks, 3
 time, 243-253
 uninitialized variables (compile-time checking), 206
 volatile modifier, 20-21
 watching, 205

VESA (Video Electronics Standards Association), 268-269
 BIOS standard, 266-267

VGA cards, 268

VGA graphics modes, changing, 265-269

Video BIOS, 265

virtual key codes (Windows), 398-400

virtual memory (page thrashing), 17-18

void pointers, 138-139
 arithmetic operations, 157
 raw memory, 138-139
 return statements, 166
 type casts, 26

volatile declarations, 20-21
 const variables, 21
 type casts, 27

W

WaitMessage() function, 357

warm boots (Ctrl-Alt-Delete), disabling, 372-374

watches, setting, 205

wcstombs() function, 241

wctomb() function, 241

while loops (infinite loops), 200

white space (program efficiency), 334-336

Whitesmiths brace style, 339

Win32s (32-bit compilers), 356

WINDIR environment variable, 256

Windows, 385-414
 animated bitmaps, 403
 buttons, 407-408
 CallWindowProc() function, 409

carets compared to cursors, 401-402
 child window controls, 407
 command-line parameters, 350
 compiler compatibility, 394-395
 controls, 407-408
 cooperative multitasking, 357
 CreateDialog() function, 414
 CreateWindow() function, 388
 edit class, 410-411
 listbox class, 411-412
 date and time, 404
 dead keys, 400-401
 DefWindowProc() function, 402
 delay timers, 387
 device contexts, 386
 DialogBox() function, 414
 DLLs (dynamic link libraries), 393
 edit class, 410-411
 editing position, 401-402
 environment variables (WINDIR), 256
 FAR PASCAL declarations, 392
 filter functions (interrupting programs), 357-360
 fixed segments, 412-413
 functions compared to DOS, 392-393
 GDI (Graphic Device Interface), 389-390
 GetKeyState() function, 400
 GetSysColor() function, 405-406
 GetWindowLong() function, 409
 global heap, 412
 GlobalAlloc() function, 413-414
 GlobalFree() function, 414
 GlobalLock() function, 414
 GlobalUnlock() function, 414
 HANDLE, 394
 handles, 387-388
 HDC, 394

heaps, 412-413
 HWND, 394
 interrupting programs, 357-360, 388-389
 KillTimer() function, 387
 listbox class, 411-412
 localtime() function, 404
 memory
 allocating, 413
 organizing, 412-413
 modal and modeless dialog boxes, 414
 mouse
 clicks, capturing, 402-403
 positioning, 401-402
 movable segments, 412-413
 OEM (Original Equipment Manufacturer) key codes, 397-398
 pascal calling convention, 392
 printf() function, 386
 repainting client areas, 395-396
 resource editors (buttons and controls), 407
 SDK (Software Development Kit), 391-392
 SetSysColor() function, 406
 SetTimer() function, 387
 SetWindowLong() function, 409
 SetWindowText() function, 405
 sizing client areas, 396-397
 sprintf() function, 386
 static child windows, 408
 subclassing windows, 409
 system colors, 405-407
 time() function, 404
 title bars, updating, 405
 virtual key codes, 398-400
 windows, saving and refreshing, 395-396
 windows.h, 390-391
 WINSTUB.EXE, 391
 WM_PAINT message, 395-396
 WM_SIZE message, 396-397

windows
 HWND (handle), 394

refreshing, 395-396
 saving, 395-396
 subclassing, 409
windows.h, 344, 390-391
WINSTUB.EXE (Windows SDK), 391
WM_PAINT message (Windows), 395-396
WM_SIZE message (Windows), 396-397
writing
 data to the screen, 295-296
 programs, 315-329
 text to the screen, 296-300
wsprintf() function, 404

X-Y-Z

x=y=z (operator precedence), 287
XMS (extended memory), 326
zero bits and calloc(), 149
zero-padding numbers, 307