

# Index

## SYMBOLS & NUMERICS

**&&** (and) operator, 171  
**'** (apostrophe) in VBScript, 98, 99, 100  
**=** (assign value to variable), 235  
**==** (check variable against value), 235  
**\*/** (close multiline comment), 102, 103  
**-** (collapse item), 342  
**{}** (curly braces), 171, 237  
**\\** (double backslash), 355  
**+** (expand item), 342  
**//** (forward slashes)  
   in C++, 103  
   in JavaScript, 101  
**<<** (insertion operator), 150, 151  
**\*** (multiplication) operator, 202  
**!=** (not equal to) operator, 171  
**/\*** (open multiline comment), 102, 103  
**||** (or) operator, 171  
**;** (semicolon)  
   in C++, 150, 151, 238–239  
   in JScript, 354  
**\_** (underscore)  
   in name of file, 369  
   in name of variable, 121, 122–123  
**7-Zip** compression tool, 443

## A

**academic use, limits on**, 92  
**Active Server Pages (ASP)**, 22  
**addictive, programming as**, 24  
**Additions/Changes Checklist**, 289  
**airflow**, 73  
**Allen, Paul** (programmer), 3  
**alphanumeric character, variable to hold**, 110  
**ambiguity, removing**  
   from code, 238  
   from output screen, 206–211  
**American Standard Code for Information Interchange (ASCII)**,  
   6, 57–64  
**Analytical Engine**, 1–2  
**and (&&) operator**, 171  
**annotating output**, 259–260

**apostrophe (')** in VBScript, 98, 99, 100  
**appending file**, 319–320  
**applications**. See **software applications**  
**Arch version control system**, 377  
**archiving file**, 265  
**arithmetic operators**  
   description of, 65  
   in name of variable, 119  
**arrays**  
   description of, 180–181  
   exercises, 183–184  
   multidimensional, 182–183  
   two-dimensional, 182  
**ASCII (American Standard Code for Information Interchange)**,  
   6, 57–64  
**ASP (Active Server Pages)**, 22  
**assigning value to variable (=)**, 235  
**assignment operators**, 66  
**Atari BASIC, "Hello, World!" code in**, 7  
**author, Web site of**, 446  
**Axialis AX-Cursors 4.5 cursor-editing and -managing program**,  
   445

## B

**Babbage, Charles (inventor)**, 1–2  
**backing up**  
   encoded script, 403  
   source code, 299  
   Windows Registry  
     in its entirety, 332–338  
     subkeys, 330–332  
     in Windows XP, 330–331  
**Backup Job Information dialog box**, 336  
**Backup or Restore Wizard**, 332, 333, 339  
**Backup Progress dialog box**, 338  
**BASIC**  
   "Hello, World!" code in, 7  
   Web sites related to, 439–440  
**beta release**, 297–298  
**binary**  
   bit grouping, 47–50  
   description of, 44  
   interpreting, 44–46

## binary (continued)

large numbers, 46–47  
purpose of, 54

## binary data, 40

## binary language, 5

## binary math

overview of, 50–51  
Windows Calculator, using, 51–54

## bit, 44, 47

## bit grouping, 47–50

## B-language, 3

## Boole, Charles (symbolic logic system inventor), 2

## Boolean value, as variable, 106

## Borland C++ compiler

in action, 93–94  
Digital Mars compared to, 381–386  
downloading and installing, 92–93  
error handling, 388–391

## breaking up code, 31–34

## browser

testing JavaScript examples in, 108  
as tool, 88

## buffer underrun protection feature, 413

## burning disc

CD/DVD for, 419–420  
drive for, 412–413  
labeling and, 420  
requirements for, 412  
software for, 414–419

## Burnproof feature, 413

## Bush, Vannevar (inventor), 2

## button. See also radio button

description of, 268–269  
example of, 277–278, 280–281

## byte, 48

## bytecode, 89, 393

# C

## C++ Builder, 232

## C programming language

commercial programming and, 20  
development of, 3  
“Hello, World!” code in, 7

## C++ programming language

advantages of, 79  
arrays, 180–183  
calculator, creating, 163–171  
code  
for temperature conversion program, 196–204  
for temperature conversion program, improving, 204–212

code structure, 138–141

commenting code, 103–104

commercial programming and, 20–21

compiling code, 141–149

conditionals, 157–171

description of, 91–92

functions, 36, 38, 141, 149–156

“Hello, World!” code in, 7, 10

IF statement, 158

logic errors, 235–236

loops, 174–178

overview of, 138

runtime errors, 230–232

statement in, 35

text-based interface, 251–254

Web sites related to, 437–439

## calc() function, 264–265

## calculator application

annotating output, 259–260  
breaking problem into smaller steps, 194–195  
coding phase, 196–204  
confirmations, 267  
confirming exit from, 260–261  
creating, 163–171  
help, adding, 261–266  
improving code, 204–212  
program overview, including, 254–256  
prompting user for inputs, 256–259  
requirements for, 186–191  
research for, 191–194

## calling

code, 34  
compiler, 144  
function, 154–156

## capitalization scheme for naming variables, 122

## CD

advantages of, 410  
burning, 412–419  
description of, 409  
disadvantages of, 410–411

## cd (change directory) command, 147

## chair, 73, 360

## change directory (cd) command, 147

## char variable, 164–167

## character

alphanumeric, variable to hold, 110  
ASCII, 57–60  
description of, 57  
extended character set, 60–62  
keyboard input and, 167–168  
in string, 124–125

## cheap disc, using for burning, 419

## checkbox

description of, 271–272  
example of, 282

## checking

for duplicate files, 317  
preceding statements to spot errors, 237  
variable against value (==), 235

## cin instruction, 161, 165–166

## client requirements

clarifying, 186–191  
reviewing, 211–212

## client-side programming, 22–23

## closing

HTML file, 108  
multiline comment (\*//), 102, 103

## COBOL

development of, 3  
“Hello, World!” code in, 8

## code. See also programming stage of project; structure of code

calling, 34  
commenting  
C++, 103–104  
encoding and, 403  
exercises, 100–101, 104  
JavaScript, 101–103  
overview of, 97–98  
programming stage of project and, 294–296  
spotting errors and, 237

- temperature conversion program, 198
- VBScript, 98–101
- compiled
  - alternatives to, 397–403
  - description of, 41, 42
  - runtime errors and, 230–232
  - version control and, 376–377
- compiler, 11
- compiling
  - benefits of, 394–397
  - C++, 141–149
  - debugging and, 397
  - functionality and, 396
  - intellectual property protection and, 395
  - process for, 379–381
  - security and, 396–397
  - speed and, 395–396
  - spotting errors and, 226
  - temperature conversion program, 203, 211
- copying, 42
- development environment and, 11
- in different languages, 6–10
- functions and procedures, 36–37, 38–39
- improving for temperature conversion program, 204–212
- inefficiency in, 285
- JavaScript
  - to display text string, 125
  - processing inputs, 130–132
  - string manipulation, 126–130
  - variable manipulation, 132–135
  - variables in action, 109–118
- logic of, 198–199
- making hard to follow, 397–403
- reading
  - breaking up code, 31–34
  - overview of, 27
  - top-down, 28–30
- reusing, 32–33
- saving, 299
- statement, 35–36, 37–38
- storage of, 39–42
- syntax of, 39
- testing
  - in browser, 108
  - to spot errors, 239
- tweaking, 300
- code-writing process, 6**
- collapsing item (-), 342**
- Command Prompt, 143**
- command window**
  - opening, 143–144
  - view of, 88
- commenting code**
  - C++, 103–104
  - encoding and, 403
  - exercises, 100–101, 104
  - JavaScript, 101–103
  - overview of, 97–98
  - programming stage of project and, 294–296
  - spotting errors and, 237
  - temperature conversion program, 198
  - VBScript, 98–101
- commercial/application programming**
  - beta and preview strategy for, 298
  - description of, 20
  - free tools and, 23
- comparison operators, 66–67**
- compiled code**
  - alternatives to, 397–403
  - description of, 41, 42
  - interpreted code compared to, 395–396
  - runtime errors and, 230–232
  - version control and, 376–377
- compiler**
  - Borland C++
    - Digital Mars compared to, 381–386
    - downloading and installing, 92–94
  - calling, 144
  - description of, 11, 25
  - error handling and, 387–391
  - executable file and, 379
  - language and, 391–394
  - license agreement, 405
- compiler error**
  - description of, 214
  - in large segment of code, 221–226
  - opening curly brace missing, 217–218
  - opening parenthesis instead of curly brace, 215–217
  - semicolon missing, 220–221
  - too many < in code, 219–220
  - terminated string, 218–219
  - warnings, generating, 226–228
- compiling code**
  - benefits of, 394–397
  - C++, 141–149
  - debugging and, 397
  - functionality and, 396
  - intellectual property protection and, 395
  - process for, 379–381
  - security and, 396–397
  - speed and, 395–396
  - spotting errors and, 226
  - temperature conversion program, 203, 211
- compression tools, 83–84, 442–444**
- computer**
  - for programming, 24
  - workspace, creating, 362–363
- computer science, degree in, 24**
- concatenating strings in JavaScript, 126–130**
- conditionals**
  - choice of folder and, 310–311
  - exercises, 172–173
  - overview of, 157
- confirmations, adding to interface, 267**
- confirming exit from application, 260–261**
- converting**
  - ASCII to binary string, 63
  - binary string to ASCII, 64
- copyright issues, 423**
- copyright notice, 403**
- cost of programming tools, 23**
- Courier New font, 73**
- cout instruction**
  - cin and, 165–166
  - cin compared to, 161
  - compiler and, 381
  - description of, 150, 151
- .cpp file extension, 141**
- CreateFolder method, 310–311**
- Crimson Editor (text editor), 434**
- curly braces ({}), 171, 237**

## D

### data storage, 39–42

#### debugging. *See also* errors

- breaking up code and, 32
- compiling code and, 397
- poor planning and, 286
- reproducing error before fixing, 300
- testing after, 300
- top-down processing and, 30

### decimals, in binary, 44

### decision making, 157–158

#### declaring

- array, 181, 182
- path for file, 309
- variable, 109, 160

### defining variable, 109–110

#### deleting

- file, 325
- folder, 325
- registry keys, 354

### design errors, checking for, 297

### desk, 73

### desktop, saving files to, 363

### development environment, 11

### dialog box, example of, 277–279

### Difference Engine, 1, 2

### Digital Mars compiler

- Borland C++ compiler compared to, 381–386
- error handling, 388–391

### distraction, minimizing, 360

#### distribution

- full-time job versus hobby, 423
- license agreement and, 405
- physical, 405–412
- supported versus unsupported, 424–425
- types of, 405
- virtual, 421–424

### DJGPP compiler, 381

### Do While loop, 177–178

#### documenting

- errors, 294
- ideas, 287–289

### double backslash (\\), 355

### doubleword, 50

#### downloading

- Borland C++ compiler, 92
- Java SDK and VM, 89–90

#### drop-down menu

- description of, 275
- example of, 277–278

### DVD

- advantages and disadvantages of, 412
- burning, 412–419

### DWORD value, 344

## E

### editing. *See also* text editor

- file
  - appending file, 319–320
  - overview of, 317–319
- Windows Registry
  - in JScript, 354–355
  - overview of, 329–330
  - in VBScript, 349–354

### else statement, 162

### e-mail support, 424

### encryption compared to encoding, 399

### endl insertion operator, 150, 151, 160, 161

### ergonomic keyboard, 72

### Error Documenting, 294

### error handling and compiler, 387–391

#### error messages

- Borland C++ compiler, 388, 390
- Digital Mars compiler, 388–389
- interpreting, 216–217
- terminated string, 219

### errors. *See also* debugging

- compiler type, 214–228
- documenting, 294
- as expected, 213–214
- logic type, 232–236
- reproducing before fixing, 300
- runtime type, 228–232
- spotting
  - checking preceding statements, 237
  - commenting and, 237
  - keeping layout clear, 237
  - keeping track of variables, 239–244
  - reading each line after pressing enter, 236
  - removing ambiguity in code, 238
  - semicolons, 238–239
  - testing code and, 239
- types of, 214
- warnings compared to, 226–228

### escape characters in JavaScript and JScript, 355

#### executable file

- compiler and, 379
- .exe extension, 11, 146, 150

#### exercises

- arrays, 183–184
- commenting code
  - C++, 104
  - JavaScript, 102–103
  - VBScript, 100–101
- conditionals, 172–173
- functions, 156–157
- loops, 178–180
- variables, 124

### expanding item (+), 342

## F

### FAQ (Frequently Asked Questions), 424

### fast disc, using for burning, 420

### feature creep, 286

#### features

- adding, 299–300
- missing, 286, 297

#### file

- appending, 319–320
- checking for duplicate, 317
- .cpp extension, 141
- creating, 306–309
- deleting, 325
- editing, 317–319
- .exe (executable), 11, 146, 150
- executable, and compiler, 379
- .ini (initialization), 328
- iostream.h, 139
- .java extension, 391

- .js extension, 355
  - .jse extension, 403
  - multiple, creating, 311
  - naming conventions, 368–370
  - opening for reading, 320
  - organizing
    - folder contents note, 366–368
    - language, grouping by, 364–365
    - overview of, 359
    - project, grouping by, 365–366
  - prompting user for name of, 314–317
  - Read method, 323–324
  - ReadAll method, 321–322
  - reading, 321
  - ReadLine method, 322–323
  - .reg (registration entries), 339
  - saving
    - to desktop, 363
    - frequency of, 362
    - principles of, 303–304
  - size of, and compiler, 382, 383, 387, 393–394
  - string.h, 139, 151
  - .txt (text), 366
  - .vbe extension, 403
  - .vbs extension, 307
  - .zip extension, 83
  - file life cycle, 304–305**
  - find and replace feature, 397**
  - flexibility of code, 33**
  - float **variable, 164–166**
  - floating point number**
    - integer compared to, 196
    - as variable, 105–106
  - floppy disk**
    - advantages of, 406
    - disadvantages of, 408–409
    - drive, 408
    - examples of, 407
  - folder**
    - conditionals and choice of, 312–313
    - contents note, 366–368
    - creating, 310–311
    - creating file and, 309
    - deleting, 325
    - grouping
      - by language, 364–365
      - by project, 365–366
    - prompting user for name of, 314–317
    - variables and, 313
  - fonts, 73–75**
  - FOR loop, 174–176**
  - formatting message box, 244**
  - FORTRAN language**
    - development of, 3
    - “Hello, World!” code in, 8
  - forum, support using, 424–425**
  - forward slashes (/ /)**
    - in C++, 103
    - in JavaScript, 101
  - free software, beta and preview strategy for, 298**
  - freelance programmer, 14**
  - freeware, 3**
  - Frequently Asked Questions (FAQ), 424**
  - functions**
    - C++, 36, 38, 141, 149–156
    - calling, 154–156
    - commenting, 296
    - defining, 140
    - description of, 36–37, 38–39
    - exercises, 156–157
    - main(), 150, 152–156, 263–264
    - overview of, 149–156
    - planning project and, 292
    - removing unused, 300
- ## G
- Gates, Bill (programmer), 3**
  - general tools**
    - Snagit, 86–87
    - UltraCompare, 84–86
    - Windows Notepad, 80–82
    - Winzip, 83–84
  - GNU General Public License, 423**
  - GNU Revision Control System, 377**
  - grouping folders**
    - by language, 364–365
    - by project, 365–366
- ## H
- halfword, 48–49**
  - hard drive, storage capacity of, 40**
  - hardware and Windows Registry, 328**
  - header file, 139–140, 151**
  - hello function, 152–156**
  - hello2 function, 152–156**
  - “Hello, World!”**
    - code examples, 7–10
    - origin and history of, 10
  - help, adding to interface, 261–266**
  - help file, creating, 301**
  - helpsystem() function, 265–266**
  - hexadecimal number system**
    - interpreting, 55–56
    - overview of, 54–55
    - Windows Calculator and, 57
  - history of programming, 1–4, 77**
  - HJ-Install freeware installer, 444**
  - hobby, programming as, 78–79, 423**
  - hobby project, 17–18**
  - Hollerith, Herman (inventor), 2**
  - Hopper, Grace Murray (programmer), 3**
  - HTML comment tags, 100, 107**
  - Hungarian Notation, 123–124**
- ## I
- IBM, 2, 3**
  - Icon Extractor utility, 445**
  - Icon Grabber utility, 445**
  - IDE, 25**
  - idea**
    - documenting, 287–289
    - maturing time for, 289
    - types of, 286–287
  - If statement**
    - description of, 158
    - temperature conversion program, 199, 207–211
  - infinite loop, 176, 234–245**
  - .ini (initialization) file, 328**
  - Initial Checklist, 287–289**

**initial instruction for use for temperature conversion program, 204–205**

**initialization file (.ini), 328**

**initializing variable, 105**

**inputs**

- buttons and, 268–269
- folder and filenames, 314–317
- processing, 130–132
- prompting user for, 256–259
- from user for temperature conversion program, 188, 195, 196–197

**insertion operator**

- <<, 150, 151
- endl, 150, 151, 160, 161

**installing**

- Borland C++ compiler, 92–94
- Digital Mars compiler, 381
- Java SDK and VM, 90–91
- Microsoft Script Encoder, 399

**integer. See also number**

- code interpreting as string, 134–135
- floating-point number compared to, 196
- replacing with text string, 114
- as variable, 106

**integrating variables with test output, 209–210**

**intellectual property, protection of, 395**

**interface**

- annotating output, 259–260
- appalling, 250–251
- buttons, 268–269
- check boxes, 271–272
- confirmations, 267
- confirming exit from application, 260–261
- description of, 247, 250
- drop-down menus, 275
- goal of, 275
- help, adding, 261–266
- importance of, 251
- menus, 270–271
- Microsoft OneNote 2003, 280–282
- Microsoft Word, 247–250
- multiline text boxes, 274
- problem solving and, 195
- program overview, including, 254–256
- prompting for input, 256–259
- radio buttons, 272–273
- simple, 250
- single-line text box, 273–274
- text-based, 251–254
- Windows Notepad, 276–280

**Internet. See also Web sites**

- virtual distribution and, 421–424
- Web programming and, 21–22

**Internet Explorer**

- testing JavaScript examples in, 108
- as tool, 88

**interpreted code compared to compiled code, 395–396**

**interpreted language and runtime errors, 228, 232**

**interpreter, 6, 11**

**interpreting**

- binary, 44–46
- error messages, 216–217
- hexadecimal number system, 55–56

**iostream.h file, 139**

## J

**Java**

- advantages of, 79
- commercial programming and, 20, 21
- compiler and, 391–394
- description of, 89
- “Hello, World!” code in, 8
- Software Development Kit (SDK), 89
- Web sites related to, 435–437
- .java file extension, 391

**Java Virtual Machine (VM) runtime environment, 89, 393, 394**

**JavaScript**

- advantages of, 79
- client-side programming and, 23
- code
  - to display text string, 125
  - processing inputs, 130–132
  - string manipulation, 126–130
  - variable manipulation, 132–135
  - variables in action, 109–118
- commenting code, 101–103
- escape characters in, 355
- “Hello, World!” code in, 9, 10
- parentheses in, 355
- plain text and, 41
- template, creating, 107–108
- tools needed for, 94–95

**JBuilder Foundation IDE/compiler, 435**

**JCreator IDE, 435**

**JEdit text editor, 434–435**

**JediVCS (Version Control System), 377**

**.js file extension, 355**

**JScript (Microsoft)**

- description of, 349
- Registry editing, 354–355

**.jse file extension, 403**

## K

**keeping code layout clear, 237**

**keeping track of variables, 239–244**

**key, finding, 341–343**

**keyboard**

- buttons and, 268–269
- choosing, 71–73
- custom, 362
- height of, 360
- input from, 167–168
- menu and, 271

**keyword**

- in name of variable, 120
- var, 109, 117–118
- void, 140

## L

**labeling**

- button, 269
- check box, 272
- disc, 420–421
- radio button, 273
- text box, 274

**language**

- changes in, 25

- choosing
    - to learn, 75–76
    - for project, 293
  - code and, 6–10
  - compiler and, 391–394
  - of computer, 4–5
  - grouping folders by, 364–365
  - knowledge of as transferable, 25
  - platform-independent, 89
  - scripting, tools needed for, 94–95
  - Web scripting sites, 441
- layout**
- for folder contents note, 366–367
  - of Windows Registry, 329
- learning programming**
- career programmer and, 14–16
  - foreign language learning and, 76
  - for fun, 18
  - as hobby, 78–79
  - knowledge as transferable to other language, 25
  - to make contribution, 18
  - overview of book, 79–80
  - problem solving and, 16–17
  - reasons for, 13–14
  - school/college setting, 77
  - time requirement, 24
  - work-based setting, 77–78
- learning/academic programming, 21**
- legal advice, 423**
- license agreements, 405**
- licensing software, 423**
- life cycle of file, 304–305**
- lighting, 73**
- line break character, 244**
- line numbering and text editor, 232**
- logic errors**
- description of, 232–236
  - testing for, 296
- logical operators, 67–68**
- loops**
- description of, 235
  - Do While type, 177–178
  - exercises, 178–180
  - For type, 174–176
  - infinite, 176, 234–245
  - semicolon and, 238–239
  - While type, 176–177
- M**
- main() function**
- calculator application, 263–264
  - description of, 150
  - example of, 152–156
- manipulating**
- stored data, 114–115
  - strings, 126–130
  - variables, 132–135
- Mathematica, “Hello, World!” code in, 9**
- Mead, Ian (programmer), 16**
- memory**
- breaking up code and, 33
  - top-down processing and, 28–29
- menu**
- description of, 270–271, 275
  - example of, 276, 277–278, 280
- message box, formatting, 244**
- methods**
- CreateFolder, 310–311
  - Read, 323–324
  - ReadAll, 321–322
  - ReadLine, 322–323
  - RegDelete, 353–354
  - RegRead, 352–353
  - RegWrite, 351
- Microsoft**
- Developer Network Web site, 446
  - JScript
    - description of, 349
    - Registry editing, 354–355
  - OneNote 2003 interface, 280–282
  - Script Encoder, 398–403
  - VBScript
    - advantages of, 79
    - appending file, 319–320
    - checking for duplicate files, 317
    - client-side programming and, 23
    - commenting code, 98–101
    - creating file, 307–309
    - creating folder, 310–311
    - deleting file, 325
    - deleting folder, 325
    - editing file, 317–319
    - folder, choice of and conditionals, 312–313
    - folder, variables and, 313
    - haphazard example of, 228–229
    - logic error, 232–234
    - multiple files, creating, 311
    - opening file for reading, 320
    - plain text and, 41
    - prompting for file and folder name, 314–317
    - reading file, 321–324
    - Registry editing, 349–354
    - runtime errors, 229–230
    - tools needed for, 94–95
    - working with files and, 306
  - Visual Basic
    - advantages to beginner of, 21
    - commercial programming and, 20–21
    - function in, 37
    - “Hello, World!” code in, 9
    - statement in, 35, 37
  - Windows
    - desktop, saving files to, 363
    - profile, creating, 362–363
    - Properties screen, Summary tab, 373–375
    - Search feature, 371–372
  - Windows Registry
    - backing up in Windows XP, 330–338
    - description of, 327
    - editing in JScript, 354–355
    - editing in VBScript, 349–354
    - file creation and, 306
    - finding subtree, key, subkey, or value, 341–343
    - information in, 328–329
    - layout of, 329
    - Regedit and Regedit32, 329–330
    - restoring, 339–340
    - subkey, adding, 343–344
    - uses for, 355–357
    - value, adding, 344–346
    - value, changing existing, 346–347

## Microsoft (continued)

- value or subkey, deleting, 348–349
- value or subkey, renaming, 347–348
- Windows Script Host, 306, 307, 357
- Word
  - file, saving, 303–304
  - interface, 247–250

## minimizing distraction, 360

## mixed case, in name of variable, 121

## modifying requirements of project, 292

## monitor

- positioning, 73
- size of, 361

## mouse

- buttons and, 268
- custom, 362
- menu and, 271

## multidimensional array, 182–183

## multiline comment

- in C++, 103
- in JavaScript, 101–102

## multiline text box, 274

## multiplication (\*) operator, 202

## MWSnap utility, 445

## myths and facts, 23–25

# N

## naming

- files, conventions for, 368–370
- variables
  - capitalization scheme, 122
  - clearer scheme for, 121–122
  - dos and don'ts for, 118–121
  - Hungarian Notation, 123–124
  - underscore and, 122–123

## Nero (Ahead Software) CD creation tool, 414–419, 441

## Netbeans IDE, 436

## Netscape. See JavaScript

## not equal to (!=) operator, 171

## Notepad text editor (Windows)

- description of, 15, 80–82
- interface, 276–280

## null variable, 105

## number. See also integer

- floating point, 105–106, 196
- in name of variable, 118–119, 120
- as variable, 105–106

## numerical inputs, modification for, 256–257

## nybble, 47–48

# O

## object code, 41

## OneNote 2003 (Microsoft) interface, 280–282

## opening

- file for reading, 320
- HTML file, 107
- multiline comment (/\*/), 102, 103

## open-source software, 18

## operators

- arithmetic, 65
- assignment, 66
- categories of, 65
- comparison, 66–67

logic errors and, 235–236

logical, 67–68

string, 68–69

## options window (Microsoft OneNote 2003), 281 or (|) operator, 171

## organizing

computer workspace, creating, 362–363

files

filename control, 368–370

folder contents note, 366–368

language, grouping by, 364–365

overview of, 359

project, grouping by, 365–366

version control

compiled code and, 376–377

software for, 377

Summary information and, 373–375

tombstone comment block and, 370–371

Windows Search and, 371–372

workspace, 360–362

## outputs to screen

ambiguity in, removing, 206–211

annotating, 259–260

checking variables and, 239–244

Java compiler, 394

temperature conversion program, 188, 195

# P

## packaging disc for distribution, 420–421

## parentheses in JavaScript and JScript, 355

## Pascal

development of, 3

“Hello, World!” code in, 9

## path, declaring for file, 309

## path information, storing in Windows, 148

## PC System Questions for software previewer, 298–299

## Perl, “Hello, World!” code in, 9

## Perl Hypertext Preprocessor (PHP), 22

## physical distribution. See also burning disc

CD, 409–411

DVD, 411–412

floppy disk, 406–409

labeling and, 420–421

overview of, 405–406

packaging and, 420–421

## pkzip compression tool, 443

## plain text, 41, 42

## Plankalkül programming language, 2

## planning project

help file, creating, 301

idea

documenting, 287–289

maturing time for, 289

types of, 286–287

language, choosing, 293

requirements

drawing up set of, 290–292

modifying, 292

stages of, 360

## planning time, 360

## platform-independent language, 89

## pop-up box, displaying variable values in, 110–114

## pop-up menu, 270

## preview release, 297–299

**problem solving**

- basics of, 186
- breaking problem into smaller steps, 194–195
- coding phase, 196–204
- improving code, 204–212
- learning programming and, 16–17
- overview of, 185
- requirements, clarifying, 186–191
- research and, 191–194
- reviewing client requirements, 211–212

**procedures, 36–37, 38–39****processing inputs, 130–132****processing steps of project, 292****profile, creating, 362–363****program overview, including in interface, 254–256****programming career**

- definition of, 13
- examples of, 14–16

**programming, definition of, 4****programming stage of project**

- basics of application and, 293
- commenting code, 294–296
- features, adding, 299–300
- testing
  - final stage of, 300–301
  - preview release, 297–299
  - steps for, 296–297
  - while writing code, 293–294
- tweaking code, 300

**programming tools**

- C++, 91–94
- command window, 88
- Internet Explorer, 88
- Java, 89–91
- scripting languages, 94–95
- Windows Explorer, 87–88

**programs. See software applications****project. See also planning project; programming stage of project**

- additional features for, 299–300
- folder contents note, 366–367
- grouping folders by, 365–366
- size of, 17
- time-critical, 17
- without planning, 285–286
- work- or hobby-related, 17–18

**prompting user**

- for file and folder name, 314–317
- for input, 256–259

**Properties screen, Summary tab (Windows), 373–375****prototyping output, 207–211****punctuation, in name of variable, 119****Python, “Hello, World!” code in, 9****Q****QBASIC, “Hello, World!” code in, 9****R****radio button**

- description of, 272–273
- example of, 282

**Rapid-Q programming language, 439–440****Read method, 323–324****ReadAll method, 321–322****reading**

- code
  - breaking up code, 31–34
  - overview of, 27
  - top-down, 28–30
- each line after pressing enter to spot errors, 236
- opening file for, 320
- registry keys, 352–353

**ReadLine method, 322–323****redefining variable, 117****.reg (registration entries) file, 339****RegDelete method, 353–354****Regedit and Regedit32, 329–330****registering copyright, 423****registration entries (.reg) file, 339****RegRead method, 352–353****RegWrite method, 351****REM statement (VBScript), 98–101****removing**

- ambiguity
  - from code, 238
  - from output screen, 206–211
- unused functions from code, 300

**representing characters, 57–64****reproducing error before fixing, 300****requirements**

- of client
  - clarifying, 186–191
  - reviewing, 211–212
- of project
  - drawing up set of, 290–292
  - modifying, 292

**rerecordable disc, using for burning, 420****research**

- maturing time for idea and, 289
- problem solving and, 191–194

**reserved word, in name of variable, 120****resources. See Web sites****restoring Windows Registry, 339–340****reusing**

- code, 32–33
- variables, 115–116

**rewritable CD, 413****Ritchie, Dennis (programmer), 3****room for work, 360****root keys, abbreviations for, 350****running code on demand, 34****runtime errors, 228–232****S****saving data**

- to desktop, 363
- file life cycle, 304–305
- frequency of, 362
- principles of, 303–304
- in Windows Registry, 343

**Scientific command, View menu (Windows Calculator), 51–52****screen resolution, 361****ScreenRip32 utility, 445****screenshot, taking, 86–87****script block, 108****Script Encoder (Microsoft), 398–403****security**

- compiling code and, 396–397
- encoding and, 398–403

**self-employment, 14****semicolon (;)**

- in C++, 150, 151, 238–239
- in JScript, 354

**server-side programming, 22****7-Zip compression tool, 443****Short Code language, 3****Simonyi, Charles (programmer), 123****single-line text box, 273–274****size**

- of file and compiler
  - Borland C++ compared to Digital Mars, 382, 383, 387
  - Java, 393–394
- of font, 74–75

**Smalltalk**

- development of, 3
- “Hello, World!” code in, 9

**Snagit, 86–87****software applications. See also calculator application; interface; planning project; programming stage of project**

- academic use, limits on, 92
- burning CD/DVD, 414–419
- “learning edition” copies of, 80
- licenses for, 433
- licensing, 423
- for version control, 377
- Windows Registry and, 328

**software maker, niche for, 14–15****source code. See code****SourceJammer version control system, 377****space, in name of variable, 120****speed**

- CD/DVD burner and, 413
- compiling code and, 395–396
- virtual distribution and, 422

**spotting errors**

- checking preceding statements, 237
- commenting and, 237
- keeping layout clear, 237
- keeping track of variables, 239–244
- reading each line after pressing enter, 236
- removing ambiguity in code, 238
- semicolons, 238–239
- testing code and, 239

**SQL (Structured Query Language), 22****statement**

- checking preceding to spot errors, 237
- description of, 35–36, 37–38
- else, 162
- If
  - description of, 158
  - in temperature conversion program, 199, 207–211
- REM, 99
- semicolon and, 238–239

**storage capacity, 40****storing path information in Windows, 148****streamlining code, 300****string operators, 68–69****string.h file, 139, 151****strings**

- code interpreting integer as, 134–135
- description of, 124–125
- manipulating, 126–130

**strType value, data types for, 351****structure of code**

- arrays, 180–183
- benefits of, 138
- conditionals, 157–171
- functions, 149–156
- loops, 174–178
- purpose of, 137–138

**Structured Query Language (SQL), 22****Stuffit compression tool, 444****subkey**

- adding, 343–344
- backing up, 330–332
- creating, 350
- deleting, 348–349
- finding, 341–343
- renaming, 347–348
- restoring, 339

**subtree, finding, 341–343****Sun Java development kit, 436****support for software, 424–425****syntax**

- definition of, 39
- Microsoft Script Encoder, 400–401

**syntax error, 216****syntax highlighting feature, 75, 76****system state data, 334**

## T

**Tab key, 279–280****tab order, 280****tabbing, 280****TaskZip compression tool, 443****telephone support, 425****temperature conversion program**

- breaking problem into smaller steps, 194–195
- coding phase, 196–204
- improving code, 204–212
- requirements for, 186–191
- research for, 191–194

**template, creating**

- C++, 196
- JavaScript, 107–108
- spotting errors and, 237

**test string, 105****testbed application, 296****testing**

- after debugging, 300
- beta release, 297–298
- code
  - in browser, 108
  - to spot errors, 239
- conditional, 161–162
- encoding and, 403
- final stage of, 300–301
- logic errors and, 236
- preview release, 297–299
- principles or equations, 193–194
- during programming stage of project, 293–294
- steps for, 296–297
- temperature conversion program, 204
- top-down processing and, 30

**text box**

- multiline, 274
- single-line, 273–274

**text display, for temperature conversion program, 204**

**text editor**

- development environment and, 11
- find and replace feature, 397
- line numbering and, 232
- UltraEdit, 15–16, 75, 76, 82–83, 433–434
- Web sites for, 433–435
- Windows Notepad, 15, 80–82, 276–280

**text string, replacing integer value with, 114**

**text-based interface**

- annotating output, 259–260
- confirmations, 267
- confirming exit from application, 260–261
- description of, 251–254
- help, adding, 261–266
- program overview, including, 254–256
- prompting for input, 256–259

**Textpad text editor, 434**

**time**

- planning, 360
- for processing and top-down method, 30
- for project and planning, 286

**time-critical project, 17**

**TinyBASIC, 3**

**tombstone comment block**

- C++, 103
- description of, 295
- JavaScript, 101–102
- VBScript, 99
- version control and, 370–371

**tools**

- BASIC, 439–440
- compression, 442–444
- general
  - Snagit, 86–87
  - UltraCompare, 84–86
  - UltraEdit, 82–83
  - Windows Notepad, 80–82
  - Winzip, 83–84
- Java, 435–436
- miscellaneous, 444–445
- programming
  - C++, 91–94, 437–438
  - command window, 88
  - Internet Explorer, 88
  - Java, 89–91
  - scripting languages, 94–95
  - Web sites for, 433–435
  - Windows Explorer, 87–88

**ToolTip, 281**

**top-down processing, 28–29**

**TortoiseCVS version control system, 377**

**traditional programming, 19–21**

**tweaking code, 300**

**two-dimensional array, 182**

**.txt (text file), 366**

## U

**Ultimate++ IDE, 437**

**UltimateZip compression tool, 443**

**UltraCompare, 84–86, 384**

**UltraEdit text editor**

- description of, 15–16, 82–83, 433–434
- syntax highlighting feature, 75, 76

**underscore ( \_ )**

- in name of file, 369
- in name of variable, 121, 122–123

**unterminated string, 219**

**updating variable, 106–107**

**user information and Windows Registry, 328**

**user, prompting**

- for file and folder name, 314–317
- for input, 256–259

## V

**value**

- adding, 344–346
  - changing existing, 346–347
  - deleting, 348–349
  - finding, 341–343
  - renaming, 347–348
  - strType, data types for, 351
- var keyword, 109, 117–118**
- variables**
- char, 164–167
  - code for variables in action, 109–118
  - commenting, 296
  - declaring, 109, 160
  - exercises, 124
  - float, 164–166
  - folder name and, 313
  - integrating with test output, 209–210
  - keeping track of, 239–244
  - manipulating, 132–135
  - manipulating strings, 126–130
  - naming
    - capitalization scheme, 122
    - clearer scheme for, 121–122
    - dos and don'ts for, 66, 118–121
    - Hungarian Notation, 123–124
    - underscore and, 122–123
  - obscure names for, 397–398
  - overview of, 104–107
  - processing inputs and, 130–132
  - strings and, 124–125
  - for temperature conversion program, 196–197, 200

**.vbe file extension, 403**

**.vbs file extension, 307**

**VBScript (Microsoft)**

- advantages of, 79
- checking for duplicate files, 317
- client-side programming and, 23
- commenting code, 98–101
- file
  - appending, 319–320
  - creating, 307–309
  - deleting, 325
  - editing, 317–319
  - opening for reading, 320
  - reading, 321–324
- folder
  - choice of and conditionals, 312–313
  - creating, 310–311
  - deleting, 325
  - variables and, 313
- haphazard example of, 228–229
- logic error, 232–234
- multiple files, creating, 311

# VBScript (Microsoft) (continued)

---

## **VBScript (Microsoft) (continued)**

- plain text and, 41
- prompting for file and folder name, 314–317
- Registry editing, 349–354
- runtime errors, 229–230
- tools needed for, 94–95
- working with files and, 306

## **version control**

- compiled code and, 376–377
- software for, 377
- Summary information and, 373–375
- tombstone comment block and, 370–371
- Windows Search and, 371–372

## **View menu commands, Scientific (Windows Calculator), 51–52**

## **virtual distribution**

- cons of, 422
- free versus commercial, 422–423
- pros of, 421–422

## **Visual Basic (Microsoft)**

- advantages to beginner of, 21
- commercial programming and, 20–21
- function in, 37
- “Hello, World!” code in, 9
- statement in, 35, 37

`void` keyword, 140

**VRML, “Hello, World!” code in, 10**

# W

**warnings, dealing with, 226–228**

**Web programming, 21–23**

**Web scripting languages, 441**

## **Web sites**

- author, 446
- BASIC tools, 439–440
- Borland C++ compiler, 92–93
- C++ tools, 437–438
- CD burning, 441
- compilers, 381
- compression tools, 442–444
- distributing free software, 423
- Java, 89–90, 435–437
- Java tools, 435–436
- Microsoft
  - Developer Network, 446
  - Script Encoder, 399
  - Visual Basic, 21
- peripherals, 362
- programming tools, 433–435
- registering copyright, 423
- resources, miscellaneous, 446
- software version control, 377
- tools, miscellaneous, 444–445
- UltraCompare, 86
- UltraEdit text editor, 15
- virtual distribution, 423
- Web scripting languages, 441
- Winzip, 84

**Web support, 424–425**

**While loop, 176–177**

**whitespace in code, 298**

**Winace compression tool, 442**

## **Windows (Microsoft)**

- Calculator
  - binary and, 51–54
  - hexadecimal system and, 57
- desktop, saving files to, 363
- Explorer, 87–88
- Notepad text editor
  - description of, 15, 80–82
  - interface, 276–280
- profile, creating, 362–363
- Properties screen, Summary tab, 373–375
- Registry
  - backing up in Windows XP 330–338
  - description of, 327
  - editing in JScript, 354–355
  - editing in VBScript, 349–354
  - file creation and, 306
  - finding subtree, key, subkey, or value, 341–343
  - information in, 328–329
  - layout of, 329
  - Regedit and Regedit32, 329–330
  - restoring, 339–340
  - subkey, adding, 343–344
  - uses for, 355–357
  - value, adding, 344–346
  - value, changing existing, 346–347
  - value or subkey, deleting, 348–349
  - value or subkey, renaming, 347–348
- Script Host, 306, 307, 357
- Search feature, 371–372
- storing path information in, 148

**Winzip (compression tool), 83–84, 442**

**word, 49–50**

## **Word (Microsoft)**

- file, saving, 303–304
- interface, 247–250

**word processor file, saving, 303–304**

**work project, 17–18**

**work-based training, 77–78**

## **workspace**

- chair, lighting, and air flow, 73
- desk, 73
- digital, creating, 362–363
- fonts, 73–75
- keyboard, 71–73
- monitor, 73
- organizing, 360–362

# Y

**YABasic interpreter, 440**

# Z

**.zip file, 83**

**Zope server, 441**

**Zuse, Konrad (scientist), 2–3**