

Index

#

- `+=` (accumulative addition) operator, 17–18, 89–90
- `/=` (accumulative division) operator, 89–90
- `*=` (accumulative multiplication) operator, 89–90
- `-=` (accumulative subtraction) operator, 89–90
- `+` (addition) operator, 2, 28, 88, 91
- `+` (concatenation) operator, 2, 75–76
- `/` (division) operator, 2, 28, 88, 91
- `=` (equal) operator, 2, 25
- `**` (exponentiation) operator, 2, 28–29, 90, 91
- `>` (greater than) operator, 2
- `>=` (greater than or equal) operator, 2
- `<` (less than) operator, 2
- `<=` (less than or equal) operator, 2
- `*` (multiplication) operator, 2, 28, 88, 91
- `<>` (not equal) operator, 2
- `-` (subtraction) operator, 2, 28, 88, 91

A

- accumulation operators, 88–90
- Add (Add) operation, 4, 88
- Adddur (Add duration) operation, 82
- `%Addr` (Get address of variable) function, 101
- alternate display format, 10
- application programming interfaces (APIs), 105–107. *See also* individual APIs
- assignment statements, 25, 65–69, 87–92
- asterisk (*), 10

B

- beginning-of-file condition, 38
- `%Bin` (Binary) function, 2
- bound call, 99–100
- built-in functions, 2–3, 18, 69–82. *See also* individual functions
- best practices for, 82
- combining, 81–82

C

- C, 2, 10, 12–13, 14, 18, 33, 35, 81, 94, 95, 102
- call and return, 97–107
- Call (Call a program) operation, 97
- CallB (Call a bound procedure) operation, 97
- CallP (Call a Prototyped Procedure or Program) operation, 29, 97–98, 101, 102, 105
- Cas (Conditionally invoke subroutine) operation, 59
 - free-format alternative to, 59, 116–117
- Cat (Concatenate two strings) operation, 75, 109
- CEEHDLR (Register a User-Written Condition Handler) API, 106, 121
- CEEHDLU (Unregister a User-Written Condition Handler) API, 106, 121
- Chain (Random Retrieval from a File) operation, 2, 17, 32, 37, 38, 39, 40, 41, 43
- `%Char` (Convert to character data) function, 70–71, 83, 96, 115
- character-to-integer conversion, 76–77
- character-to-packed decimal conversion, 71–72

- Check (Check characters) operation, 39, 79
- %Check (Check characters) function, 21, 79–80
- CheckR (Check reverse) operation, 39, 79
- %CheckR (Check reverse) function, 79–80
- CHGPRTF (Change Printer File)
 - command, 47
- CHGVAR (Change Variable)
 - command, 25
- Cobol, 10, 18, 33
- comments, 7, 8, 10–11, 29, 35
- Comp (Compare) operation, 26
- comparison symbols, 2
- concatenation, 2, 75–76
- conditional logic, 49–52
- Const keyword, 101, 102, 103–104
- Control Language (CL), 2, 25, 33
- cross-foot, 92
- CRTDTAQ (Create Data Queue)
 - command, 107
- CVTRPGSRC (Convert RPG Source)
 - command, 31
- D**
- data area input, 42
- data manipulation functions, 69–70
- data manipulation operations, 65–85
- data type
 - conversion, 70–72, 76–77, 95–96
 - date, 3
 - floating-point, 3
 - indicator, 33
 - integer, 3
 - numeric, 95
 - object, 3
 - pointer, 3
 - time, 3
 - timestamp, 3
- database input, 37–42
- database output, 42–44
- date and time operations, 82–85
- date data type, 3
- date/time arithmetic, 84–85
- date/time-to-character conversion, 83–84
- date/time-to-decimal conversion, 83–84
- %Days (Number of days) function, 84
- %Dec (Convert to packed decimal format) function, 71–72, 83, 96
- %Dech (Convert to packed decimal format with half adjust) function, 71, 96
- decimal position, extracting, 77–78
- decimal-to-character conversion, 70–71
- %Decpos (Get number of decimal positions) function, 72, 77–78
- definition specifications, 4, 42, 55, 77, 97
- Delete (Delete record) operation, 39, 42, 43, 44
- Dev keyword, 45
- %Diff (Difference between two date, time, or timestamp values) function, 84, 85
- Div (Divide) operation, 4, 88
- Do (Do) operation, 10, 27, 53
 - free-format alternative to, 118–119
- Do operations, 53–56
- Dou (Do until) operation, 27, 29, 53, 54, 56, 57, 92
- Dow (Do while) operation, 27, 29, 53–54, 56, 57, 92

Dsply (Display message) operation, 46
dynamic call, 98–99

E

(e) operation extender, 40, 119, 120, 121

%Editc (Edit value using an edit code) function, 70–71

%Elem (Get number of elements) function, 78–79

Else (Else) operation, 26–27, 49, 50, 52, 58

Elseif (Else if) operation, 26–27, 29, 49, 50, 52, 58, 116–117

embedded SQL

advantages of, 61

fixed-format method, 61–62

free-format method, 63

/End-free compiler directive, 18

end-of-file condition, 38, 57

/End-SQL precompiler directive, 62

Enddo (End a Dou or Dow group) operation, 11, 53, 54, 57, 118

Endfor (End a For group) operation, 11, 55–56, 57

Endif (End if) operation, 26–27, 49, 50, 52, 54

Endmon (End a monitor group) operation, 120

Endsl (End select) operation, 59–59

Endsr (End subroutine) operation, 119

%Eof (End-of-file), 2, 38

%Equal (Return exact match condition) function, 38, 41–42

error handling, 40, 119–121

%Error (Return error condition) function, 38, 40–41

Eval (Evaluate expression) operation, 2, 4, 24, 25, 29, 65–69, 87, 88

and blank padding, 67

character, 67–68

explicit vs. implicit, 66

fixed- vs. free-format, 92–94

vs. Move, 69

numeric, 66–67

and truncation, 67–68

EvalR (Evaluate right) operation, 68

exception handlers, 120, 121

Exec SQL precompiler directive, 62, 63

Except (Calculation time output) operation, 42, 43, 44, 47, 48

Exfmt (Write then read) operation, 44–45, 46

exponentiation, 90–91

Export keyword, 99

expression operators, 141–142

expressions, 87–92

Exsr (Invoke subroutine) operation, 59

Extended Factor 2, 2, 19, 26, 29, 35, 81, 87, 92–94, 104

externally described files, 9

Extname keyword, 39, 98

Extpgm keyword, 98

Extproc keyword, 99

F

field naming, 8–9

%Fields (Fields to update) function, 37, 43

fixed-format RPG IV, 1–4, 18
operation codes, 24

floating-point data type, 3

For groups, 78, 79

For (For) operation, 27, 29, 53, 55–56, 57, 92, 118–119

%Found (Return found condition) function, 2, 38, 39–40, 43

four-function math, 88

/Free compiler directive, 18

free-format programming languages, 12–14

free-format RPG IV

advantages of, 7–15

accessing APIs in, 105–107

alternatives for fixed-format operations, 109–121, 197–201

calculations, 18

call and return, 97–107

code block, 18–21

coding formats, 29–31

controlled looping, 27–28

data manipulation operations, 65–85

database input, 37–42

database output, 42–44

embedded SQL operations using, 61–63

expression operators, 141–142

error handling, 119–121

field naming, 8–9

getting started with, 18

introduction of, 17

introduction to, 17–35

keyed access, 32

language comparison, 12–14

math operations, 87–96

mathematics, 28–29

modern programming style of, 12–14, 18

non-case sensitivity of, 18, 20

operation codes, 18, 21–23, 143–174

operations absent in, 59–60

and programming efficiency, 15

printer output, 47–48

program flow, 49–60

programming best practices, 7–12, 20, 29–34, 35

readability and maintainability of, 7–12

sample programs, 123–139, 175–196

variable naming, 19–20, 93–94

workstation I/O, 44–46

functions. *See* built-in functions

G

Goto (Go To) operation, 8, 11, 27, 56, 57, 59

free-format alternatives to, 59–60

H

(h) operation extender, 25

half adjust, 25, 66–67, 76–77

I

If group, 49–52

If (If) operation, 2, 10, 26–27, 29, 49–52, 53, 58, 92, 116–117

ILE RPG. *See* RPG IV

In (Retrieve a data area) operation, 42

INDARA keyword, 33

INDDS keyword, 9, 33, 54

indenting, 8, 10, 35, 51, 92

indicator data type, 33

indicator set, on condition, 68

indicators, 4, 8, 68

inline composite argument list, 39, 43

input/output, 37–48

%Int (Convert to integer format)
function, 76–77, 96
%Inth (Convert to integer format with
half adjust) function, 76–77, 96
integer data type, 3
Integrated Language Environment
(ILE), 2, 106, 120, 121
Intersystem Communications Facility
(ICF), 45
Invite keyword, 45
Iter (Iterate) operation, 11, 56, 57–58,
59

J

Java, 3, 13, 14, 33, 35, 94, 95

K

%Kds (Search arguments in data
structure) function, 17, 37, 39, 43
Kfld (Define parts of a key) operation,
17, 32, 39
Klist (Define a composite key)
operation, 17, 32, 39, 43

L

Leave (Leave a Do/For group)
operation, 11, 56, 57, 59
LeaveSr (Leave a subroutine) operation,
11, 59
%Len (Get or set length) function, 72,
77–78
length, extracting, 77–78
*Like keyword, 77, 78
LikeRec keyword, 39
Linoma Software, 31
local variable concept, 3
Lookup (Look up a table or array
element) operation, 24, 39, 110

%Lookup (Look up an array element)
function, 21, 24
loop interrupt, 11, 56–58, 59
looping operations, 27–28, 53, 54, 55–
56, 56–58
LR indicator, 4, 34, 104

M

maintenance (program), 7–9
math operations, 87–96
math symbols, 2
modern programming style, 12–14
Monitor (Begin a monitor group)
operation, 120, 121
%Months (Number of months) function,
84, 85
Move (Move) operation, 24, 69, 70
free-format alternative to, 113–115
MoveA (Move array) operation
free-format alternative to, 109–113
MoveL (Move left) operation, 67, 69,
70
free-format alternative to, 115–116
Mult (Multiply) operation, 4, 88
Mvr (Move remainder) operation, 24,
88

N

named indicator, 33, 47, 68
not found condition, 38
number-to-date/time conversion, 82

O

object data type, 3
OFLIND keyword, 34
On-error (On error) operation, 120

operation codes, 21–23, 24. *See*
individual operation names

operation extenders, 25, 40, 119, 120,
121, 143

Options(*NoPass) keyword, 101

Options(*Omit) keyword, 101

Options(*RightAdjust) keyword, 102

Options(*String) keyword, 102

Options(*Trim) keyword, 102

Options(*VarSize) keyword, 101–102

OS/400

- V3R1, 1
- V5R1, 17, 26, 33, 47
- V5R2, 17
- V7.1, 2, 18, 19, 21, 74, 98

Other (Otherwise select) operation, 58–
59

output specifications, 44

overflow indicator, 34, 47

Overlay keyword, 33

OVRPRTF (Override with Printer File)
command, 47

P

parameter passing, 13, 100–104

Parm (Identify parameters) operation,
97

parentheses, 2, 26, 91

Pascal, 10, 35

PL/1, 10, 18, 35

Plist (Identify a parameter list)
operation, 97

pointer data type, 3

precedence

- in free-format expressions, 91
- in If operations, 50–51

procedure interface (PI), 98, 99

procedure prototype (PR), 97

printer output, 47–48

program flow, 49–60

programming skills

- maintaining, 14–15
- management’s perspective on, 15
- programmer’s perspective on, 14

*PSSR subroutine, 119, 121

Q

QCMDEXC (Execute Command) API,
106

QRCVDTAQ (Receive Data Queue)
API, 106–107

QSNDDTAQ (Send Data Queue) API,
106–107

qualified data structure, 39

R

(r) precision operation extender, 25

Rational Developer for Power Systems
Software (RDp), 31

read operations, 37, 38

Read (Read a record) operation, 2, 37,
40, 41, 43, 45, 46, 57

ReadE (Read equal key) operation, 37

ReadP (Read prior record) operation, 37

ReadPE (Read prior equal) operation,
37

record locking, 43

recursion, 3

%Replace (Return integer remainder)
function, 88

%Replace (Replace character string)
function, 73–74, 78, 111, 114, 115

result decimal precision, 66

resulting indicator, 38, 40, 119

- Return (Return to caller) operation, 29, 104–105
- RPG. *See also* RPG IV, RPG/400
 cycle, 4
 free format as centerpiece of, 1
 history, 1–4
- RPG II, 26
- RPG III, 26
- RPG IV. *See also* free-format RPG IV
 field, file, and record name lengths, 3
 fixed-format, 1–4
 introduction of, 1–2
 major features of, 2–3
 as modern programming language, 3–4, 5
 other changes in, 3–4
 parameter passing, 13
 preservation of old functionality in, 4
 variable names, 20
- RPG/400, 3, 4, 8, 31
- RPGLE member type, 61
- RT indicator, 4
- S**
- sample programs
 customer data entry program, 129–134, 182–190
 customer inquiry program, 128–129, 179–182
 investment and loan utility program, 123–127, 175–179
 sales report program, 135–139, 190–196
- Scan (Scan string) operation, 3, 39, 80, 109
- %Scan (Scan for characters) function, 3, 80, 111
- %Scanrpl (Replace character string) function, 74–75, 111
- Select (Begin a Select group) operation, 26, 29, 58–59, 116–117
- semicolon (;), 19, 94
- Setgt (Set greater than) operation, 37, 39, 42
- Setll (Set lower limit) operation, 37, 39, 41–42
- Setoff (Set indicator off) operation, 24
- Seton (Set indicator on) operation, 24
- Setxx (Set xx) operation, 17, 32
- short-form math, 88–90
- size, extracting, 77–78
- %Size (Get size in bytes) function, 77–78
- slash (/) character, 18
- slash, double (//), 11, 19
- spaghetti code, 27
- SQL. *See* Structured Query Language (SQL)
- SQL communications area, 62, 63
- SQLRPGLE member type, 61, 63
- Sqrt (Square root) operation, 90
- %Sqrt (Square root) function, 90
- %Sst (Substring) CL function, 2, 72
- string translation, 80–81
- structured operations, 11
- Structured Query Language (SQL), 61–63
- Sub (Subtract) operation, 4, 88
- %Subarr (Set/Get portion of an array) function, 112–113
- %Subdt (Extract a portion of a date, time, or timestamp) function, 84, 85
- Subdur (Subtract duration) operation, 82

subprocedures, 3, 99, 104–105
Subst (Substring) operation, 3, 72, 109
%Subst (Get substring) function, 3, 72–
73, 78, 110, 114, 115
substringing, 72–73

T

Tag (Tag) operation, 27
time data type, 3
timestamp data type, 3
%Tlookup (Look up a table element)
function, 24
%Trim (Trim characters at edges)
function, 75–76, 112
%TrimL (Trim leading characters)
function, 75
%TrimR (Trim trailing characters)
function, 75
truncation, with Eval, 67–68

U

Unlock (Unlock a data area or release a
record) operation, 43
%Uns (Convert to unsigned format)
function, 76–77, 96
%Unsh (Convert to unsigned format
with half adjust) function, 76–77, 96

Update (Modify existing record)
operation, 42, 43, 44
*USA date/time format, 83

V

Value keyword, 100, 101, 103
variable names, 19–20, 93–94

W

WebSphere Development Studio, 17
When (When true then select)
operation, 26, 29, 58–59, 116–117
whole-number portion, extracting, 76
work variables, 19
workstation I/O, 44–46
WORKSTN device, 45
Write (Create new records) operation,
42, 44, 45, 46, 47

X

%Xfoot (Sum array expression
elements) function, 92
Xlate (Translate) operation, 80
%Xlate (Translate) function, 72, 80–81,
111

Y

%Years (Number of years) function, 84