

Prentice-Hall

Series in Automatic Computation

George Forsythe, editor

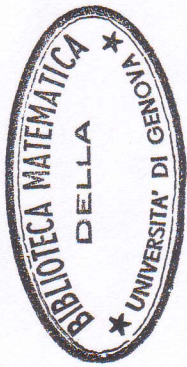
- ARBIB, *Theories of Abstract Automata*  
BATES AND DOUGLAS, *Programming Language|One*  
BAUMANN, FELICIANO, BAUER, AND SAMELSON, *Introduction to ALGOL*  
BLUMENTHAL, *Management Information Systems*  
BOBROW AND SCHWARTZ, EDITORS, *Computers and the Policy-Making Community*:  
*Applications to International Relations*  
BOWLES, EDITOR, *Computers in Humanistic Research*  
CESCHINO AND KUNTZMANN, *Numerical Solution of Initial Value Problems*  
CRESS, DIRKSEN, AND GRAHAM, *FORTRAN IV with Whatfor*  
DESMONDE, *Computers and Their Uses*  
DESMONDE, *A Conversational Graphic Data Processing System: The IBM 1130|2250*  
DESMONDE, *Real-Time Data Processing Systems: Introductory Concepts*  
EVANS, WALLACE, AND SUTHERLAND, *Simulation Using Digital Computers*  
FIKE, *Computer Evaluation of Mathematical Functions*  
FORSYTHE AND MOLER, *Computer Solution of Linear Algebraic Systems*  
GOLDEN, *FORTRAN IV: Programming and Computing*  
GOLDEN AND LEICHUS, *IBM 360: Programming and Computing*  
GREENSPAN, *Lectures on the Numerical Solution of Linear, Singular and Nonlinear Differential Equations*  
GRUENBERGER, EDITOR, *Computers and Communications—Toward a Computer Utility*  
HARTMANIS AND STEARNS, *Algebraic Structure Theory of Sequential Machines*  
HULL, *Introduction to Computing*  
LOUDEN, *Programming the IBM 1130 and 1800*  
MARTIN, *Design of Real-Time Computer Systems*  
MARTIN, *Programming Real-Time Computer Systems*  
MARTIN, *Telecommunications and the Computer*  
MINSKY, *Computation: Finite and Infinite Machines*  
MOORE, *Interval Analysis*  
SAMMET, *Programming Languages: History and Fundamentals*  
SCHULTZ, *Digital Processing: A System Orientation*  
SNYDER, *Chebyshev Methods in Numerical Approximation*  
STERLING AND POLLACK, *Introduction to Statistical Data Processing*  
STROUD AND SEACREST, *Gaussian Quadrature Formulas*  
TRAUB, *Iterative Methods for the Solution of Equations*  
VARGA, *Matrix Iterative Analysis*  
VAZSONYI, *Problem Solving by Digital Computers with PL/I Programming*  
WILKINSON, *Rounding Errors in Algebraic Processes*  
ZIEGLER, *Time-Sharing Data Processing Systems*

PRENTICE-HALL INTERNATIONAL, INC., London  
PRENTICE-HALL OF AUSTRALIA, PTY. LTD., Sydney  
PRENTICE-HALL OF CANADA, LTD., Toronto  
PRENTICE-HALL OF INDIA PRIVATE LTD., New Delhi  
PRENTICE-HALL OF JAPAN, INC., Tokyo

## PROGRAMMING LANGUAGES: History and Fundamentals

JEAN E. SAMMET

Programming Language Technology Manager  
Federal Systems Division  
IBM Corporation



68  
1969  
10

PRENTICE-HALL, INC.  
ENGLEWOOD CLIFFS, N. J.



5.2. Disadvantages	17
1. Advantages do not always exist	17
2. Time required for compiling	17
3. Inefficient object code	18
4. Difficulties in debugging without learning machine language	18
5. Inability of the language to express all needed operations	18
5.3. Overall evaluation	19
6. Classifications of Programming Languages and Proposed Definitions	19
6.1. Procedure-oriented language	19
6.2. Nonprocedural language	20
6.3. Problem-oriented language	21
6.4. Application-oriented language	21
6.5. Special purpose language	21
6.6. Problem-defining language	21
6.7. Problem-describing language	22
6.8. Problem-solving language	22
6.9. Reference language	22
6.10. Publication language	22
6.11. Hardware language	23
7. Factors in Choice of a Language	23
7.1. Suitability of language for problem area and projected users	23
7.2. Availability on desired computer	24
7.3. History and evaluation of previous use	25
7.4. Efficiency of language implementation	25
7.5. Compatibility and growth potential	25
7.6. Functional (= nontechnical characteristics)	26
7.7. Technical characteristics	26
References	26

## FUNCTIONAL CHARACTERISTICS OF PROGRAMMING LANGUAGES

1. Description of the Concept of Functional Characteristics	30
2. Properties of Languages	31
3. Purpose of Language	32
3.1. Application area	33
3.2. Type of language	34
3.3. Type of user	34
3.4. Physical environment	36
4. Conversion and Compatibility	36
4.1. Types of compatibility	37
1. Machine independence	37
2. Compiler independence	38
3. Dialects and language L-like	39
4. Subsetting and extensions	40
5. Relation to language definition	41

## CONTENTS

### GENERAL INTRODUCTION

1. Machine Language Programming	1
2. Symbolic Assembly Language Programming	2
3. Early Development of Better Tools	2
3.1. Specific needs to be met	3
3.2. Brief history of early efforts	3
4. Definition of Programming Languages	5
4.1. Definition problem	8
4.2. Defining characteristics	8
1. Machine code knowledge is unnecessary	9
2. Potential for conversion to other computers	9
3. Instruction explosion	10
4. Problem-oriented notation	10
4.3. Basic terminology	11
1. Source program	11
2. Object program	11
3. Compiler	12
4. Interpreter	12
5. Automatic coding	13
6. Automatic programming	13
4.4. Difference between programming language and application package	13
5. Advantages and Disadvantages of Higher Level Languages	14
5.1. Advantages	14
1. Ease of learning	14
2. Ease of coding and understanding	15
3. Ease of debugging	15
4. Ease of maintaining and documenting	16
5. Ease of conversion	16
6. Reduce elapsed time for problem-solving	17



4.2. Ease of conversion	41
1. Based on compatibility	41
2. Ease of SIFTing to another language	42
3. Ease of translating to another language	42
5. Standardization	43
5.1. Purposes	43
5.2. Problems	43
1. Conceptual problems	44
2. Technical problems	44
3. Procedural problems	46
5.3. Method of establishing standards	46
5.4. Overall status	48
6. Types and Methods of Language Definition	48
6.1. Administrative	49
1. Who designed the language?	49
2. What were the objectives of the language?	49
3. Who implemented the language?	50
4. Who maintains the language?	50
6.2. Technical	51
1. Syntax, semantics, and pragmatics	51
2. Formalized notation	52
6.3. Types of documentation	57
6.4. Technical	58
6.5. Evaluation Based on Use	58
7.1. Availability on differing computers	58
7.2. Evaluation of language versus evaluation of compiler	59
7.3. Usage relative to objectives	59
7.4. Advantages	60
7.5. Disadvantages	60
7.6. Mistakes to be avoided in the future	61
References	61

## TECHNICAL CHARACTERISTICS OF PROGRAMMING LANGUAGES

1. Description of Concept of Technical Features	65
1.1. Introduction	65
1.2. Major parts of language	66
1. Data and its description	66
2. Operators	66
3. Commands	67
4. Declarations	67
5. Compiler directives	68
6. Delimiters	68
7. Program structure	68
2. Form of Language	68
2.1. Character set	69
2.2. Types of basic elements (= tokens)	70
1. System-defined	70
2. User-defined and restrictions	71

2.3. Identifier definition	72
1. Types of identifiers	72
2. Formation rules	72
3. Use of reserved words	73
4. Data names for aggregates (subscripts, qualification)	74
2.4. Definition and usage of other basic elements	77
1. Operators	77
2. Delimiters	77
3. Use and meaning of punctuation	78
4. Significance of blanks	79
5. Use of noise words	79
6. Literals	80
2.5. Type of input form used	80
1. Physical input form	81
2. Conceptual form	82
3. Structure of Program	82
3.1. Types of subunits	83
1. Nonexecutable: declarations and compiler directives	83
2. Smallest executable unit	83
3. Sets of smallest executable units	84
4. Loops	85
5. Functions, subroutines and procedures	85
6. Comments	86
7. Interaction with the operating system and environment	87
8. Inclusion of other languages	87
9. Complete program (including sequencing rules)	88
3.2. Characteristics of subunits	88
1. Methods of delimiting	89
2. Recursive	89
3. Parameter passage and differing types	90
4. Embedding	92
4. Data Types and Units and Computations with Them	92
4.1. Types of data variables and constants	93
1. Arithmetic	93
2. Logical (= Boolean)	93
3. Character	93
4. Complex	93
5. Formal (= algebraic)	94
6. String	94
7. List of pointer	94
8. Hierarchical	94
9. Others	95
10. Combinations of variable and constant types	95
4.2. Accessible data units	95
1. Hardware data units	95
2. Language data units	97
4.3. Types of arithmetic	97
1. Integer, fixed point, mixed number	98
2. Floating point	98
3. Rational	98



4. Complex numbers	99
5. Double or multiple precision	99
6. Logical	99
7. Other	100
8. Higher level data units	100
4.4. Rules on creation and evaluation of arithmetic and logical expressions	100
1. Intermingling rules	100
2. Conversion rules	101
3. Precision and computation rules for various modes	102
4. Precedence and sequencing rules	102
4.5. Scope of data	103
5. Executable Statement Types	104
5.1. Assignment	104
1. Methods of specifying computation	105
2. Conversion rules for results	105
5.2. Alphanumeric data handling	106
1. Editing statements	106
2. Conversion statements	106
3. Sorting statements	106
5.3. Sequence control and decision making	107
1. Control transfer statements	107
2. Conditional statements	108
3. Loop control statements	109
4. Error condition statements	110
5.4. Symbolic data handling	111
1. Algebraic expression manipulation statements	111
2. List-handling statements	111
3. String-handling statements	112
4. Pattern-handling statements	112
5.5. Interaction with operating system and/or equipment	112
1. Input/output statements	113
2. Library reference statements	113
3. Debugging statements	114
4. Storage and segmentation allocation statements	114
5. Operating system and machine feature statements	115
5.6. Others	115
6. Declarations and Nonexecutable Statements	115
6.1. Data description	116
6.2. File description	117
6.3. Format description	118
6.4. Storage allocation	118
6.5. Environment or operating system descriptions	119
6.6. Procedure, subroutine, function declarations	119
6.7. Compiler directives	119
6.8. Others	120
7. Structure of Language and Compiler Interaction	120
7.1. Self-modification of programs	120
7.2. Self-extension of the language	121
7.3. Ability to write the compiler for a language in that language	121

7.4. Effect of language design on implementation efficiency	122
1. Compile time versus object time efficiency	122
2. Generality versus restrictions	123
3. Specific features with significant effect	124
4. Storage allocation requirements	124
5. Possibility for providing choice of tradeoffs	124
7.5. Debugging aids and error checking	125
8. Other Features Not Included	125
References	126

## IV LANGUAGES FOR NUMERICAL SCIENTIFIC PROBLEMS

1. Scope of Chapter	128
2. Languages of Historical Interest Only	128
2.1. Very early systems	129
1. SHORT CODE	129
2. Speedcoding	130
3. Laning and Zierler system	131
4. A-2 and A-3	132
5. BACAIC	133
6. PRINT	134
2.2. More widely used systems	134
1. MATH-MATIC (AT-3)	135
2. UNICODE	137
3. IT, FORTRANSIT, and GAT	139
4. ALGOL 58 (cross-reference only)	143
3. FORTRAN	143
3.1. History of FORTRAN	143
3.2. Functional characteristics of ASA (USASI) FORTRAN and Basic FORTRAN	150
3.3. Technical characteristics of ASA (USASI) Basic FORTRAN	157
3.4. Technical characteristics of ASA (USASI) FORTRAN	165
3.5. Significant contribution to technology	169
3.6. Significant extensions of FORTRAN	170
1. Proposal Writing language	170
2. FORMAC (cross-reference only)	171
3. QUIKTRAN (cross-reference only)	172
4. GRAF (cross-reference only)	172
5. DSL/90 (cross-reference only)	172
4. ALGOL	172
4.1. History of ALGOL	172
1. ALGOL 58	172
2. ALGOL 60	175
3. Revised ALGOL 60	177
4. ALGOL 6X	178
4.2. Functional characteristics of Revised ALGOL 60—Proposed ISO Standard	178
4.3. Technical characteristics of Revised ALGOL 60—Proposed ISO Standard	178



4.4. Significant contribution to technology	192
4.5. Extensions of ALGOL	194
1. Formula ALGOL (cross-reference only)	194
2. LISP 2 (cross-reference only)	195
3. AED (cross-reference only)	195
4. SFD-ALGOL (cross-reference only)	195
5. SIMULA (cross-reference only)	195
6. DIAMAG	195
7. GPL	195
8. Extended ALGOL	196
5. Languages Motivated by ALGOL 58	196
5.1. NELIAC	197
5.2. MAD	205
5.3. JOVIAL (cross-reference only)	215
6. On-Line Systems	215
6.1. Introductory remarks	215
6.2. JOSS	217
6.3. QUIKTRAN	226
6.4. BASIC	229
6.5. CPS	232
6.6. MAP	240
6.7. Lincoln Reckoner	245
6.8. APL/360 and PAT	247
6.9. Culler-Fried System	253
6.10. DIALOG	255
6.11. AMTRAN	258
7. Languages with Fairly Natural Mathematical Notation	264
7.1. Introductory remarks	264
7.2. COLASL	265
7.3. MADCAP	271
7.4. MIRFAC	281
7.5. Klerer-May system	284
8. Miscellaneous	294
8.1. CORC	294
8.2. OMNITAB	296
8.3. More nonprocedural languages	299
References	300

## V LANGUAGES FOR BUSINESS DATA PROCESSING PROBLEMS

1. Scope of Chapter	314
2. Languages of Primarily Historical Interest	316
2.1. FLOW-MATIC (and B-θ)	316
2.2. AIMACO	324
2.3. Commercial Translator	324
2.4. FACT	327
2.5. GECON	328

314

3. COBOL	330
3.1. History of COBOL	330
3.2. Functional characteristics of COBOL	334
3.3. Technical characteristics of COBOL	345
3.4. Significant contribution to technology	375
4. File Handling	376
4.1. Extensions of COBOL	376
1. IDS	376
4.2. General (cross-reference only)	376
References	377

## VI STRING AND LIST PROCESSING LANGUAGES

382

1. Scope of Chapter	382*
2. Languages of Historical Interest Only	388
3. IPL-V	388
3.1. History of IPL-V	388
3.2. Functional characteristics of IPL-V	389
3.3. Technical characteristics of IPL-V	393
3.4. Significant contribution to technology	400
4. L <sup>6</sup>	400
5. LISP 1.5	405
5.1. History of LISP 1.5	405
5.2. Functional characteristics of LISP 1.5	407
5.3. Technical characteristics of LISP 1.5	410
5.4. Significant contribution to technology	416
6. COMIT	416
6.1. History of COMIT	416
6.2. Functional characteristics of COMIT	417
6.3. Technical characteristics of COMIT	421
6.4. Significant contribution to technology	435
7. SNOBOL	436
7.1. History of SNOBOL	436
7.2. Functional characteristics of SNOBOL	436
7.3. Technical characteristics of SNOBOL	438
7.4. Significant contribution to technology	448
8. TRAC	448
9. Languages Not Widely Used	454
9.1. AMBIT	454
9.2. TREET	457
9.3. Others	461
1. CLP	461
2. CORAL	462
3. SPRINT	462
4. LOLITA	464
References	464



## VII FORMAL ALGEBRAIC MANIPULATION LANGUAGES

471

1. Scope of Chapter
2. Languages of Historical Interest Only
  - 2.1. ALGY
3. FORMAC
  - 3.1. History of FORMAC
  - 3.2. Functional characteristics of FORMAC
  - 3.3. Technical characteristics of FORMAC
  - 3.4. Significant contribution to technology
4. MATHLAB
  - 4.1. History of MATHLAB
  - 4.2. Functional characteristics of MATHLAB
  - 4.3. Technical characteristics of MATHLAB
  - 4.4. Significant contribution to technology
5. ALTRAN
6. FLAP
7. Systems Requiring Special Equipment
  - 7.1. Magic Paper
  - 7.2. Symbolic Mathematical Laboratory
- References

471  
473  
473  
474  
474  
475  
476  
490  
491  
491  
491  
493  
501  
502  
506  
510  
510  
514  
520

## VIII MULTIPURPOSE LANGUAGES

523

1. Scope of Chapter
2. Languages of Historical Interest Only
3. JOVIAL
  - 3.1. History of JOVIAL
  - 3.2. Functional characteristics of JOVIAL
  - 3.3. Technical characteristics of JOVIAL
  - 3.4. Significant contribution to technology
4. PL/I
  - 4.1. History of PL/I
  - 4.2. Functional characteristics of PL/I
  - 4.3. Technical characteristics of PL/I
  - 4.4. Significant contribution to technology
5. Formula ALGOL
6. LISP 2
- References

523  
524  
524  
524  
526  
530  
539  
540  
540  
542  
548  
582  
583  
589  
598

## IX SPECIALIZED LANGUAGES

603

1. Scope of Chapter
2. Languages for Special Application Areas
  - 2.1. Machine tool control
    1. APT
    2. Others
  - 2.2. Civil engineering
    1. COGO
    2. STRESS
    3. ICES
  - 2.3. Logical design
    1. APL (Iverson)
    2. LOTIS
    3. LDT
  4. Langage for simulating digital systems
  5. Computer Compiler
  6. Computer Design Language
  7. SFD-ALGOL
- 2.4. Digital simulation of block diagrams
  1. Introduction
  2. DYANA
  3. DYSAC
  4. DAS
  5. DSL/90
- 2.5. Compiler writing
  1. Introduction
  2. CLIP
  3. TMG
  4. COGENT
  5. META 5
  6. TRANDIR
  7. FSL
  8. AED (cross-reference only)
- 2.6. Miscellaneous
  1. Matrix computations: Matrix Compiler
  2. Cryptanalysis: OCAL
  3. Movie creation: Animated Movie Language and BUGSYS
  4. Social science research: DATA-TEXT
  5. Equipment checkout: STROBES, DIMATE
3. Specialized Languages across Application Areas
  - 3.1. Discrete simulation
    1. Introduction
    2. DYNAMO
    3. GPSS
    4. SIMSCRIPT
    5. SOL

603  
605  
605  
605  
606  
610  
611  
612  
615  
620  
620  
621  
621  
622  
623  
623  
623  
627  
625  
628  
629  
631  
632  
633  
633  
635  
636  
638  
638  
640  
641  
641  
642  
642  
642  
644  
646  
647  
650  
650  
650  
651  
653  
655  
656



6. MILITRAN	657
7. SIMULA	660
8. OPS	662
3.2. Query	662
1. Introduction	664
2. COLINGO and C-10	665
3. 473L Query	667
4. ADAM	668
5. BASEBALL	668
6. DEACON	669
7. Protosynthes	670
8. AESOP	674
3.3. Graphic and on-line display languages	674
1. GRAF	677
2. PENCIL	677
3. Graphic	678
4. DOCUS	678
5. AESOP (cross-reference only)	678
4. Computer-aided design	678
1. General	680
2. AED	684
3.5. Text editing and processing	687
3.6. Control languages for on-line and operating systems	693
References	

## X SIGNIFICANT UNIMPLEMENTED CONCEPTS

1. Scope of Chapter	707
2. UNCOL	708
3. Information Algebra	709
4. APL (Iverson)	712
5. English	715
6. Hardware Implementation of Programming Languages	717
References	719

## XI FUTURE LONG-RANGE DEVELOPMENTS

1. Introduction	722
2. Theory-Oriented Category	723
2.1. Language definition, translation, and creation	723
2.2. Next major conceptual step	725
2.3. Nonprocedural languages	726
2.4. Problem-describing languages	726
2.5. Use of mathematical concepts	727

3. User-Oriented category	727
3.1. User-defined languages	727
3.2. Use of natural language	729
3.3. Communication with hardware and software	732
3.4. Languages for new application areas	733
3.5. Languages for writing software	734
4. Interrelationships among Some of These Concepts	734
5. Conclusions and Summary	736
References	736

## A BIBLIOGRAPHY ARRANGEMENTS AND AUTHOR LIST

738

## B LANGUAGE SUMMARY

753

## NAME AND SYSTEMS INDEX

765

## SUBJECT INDEX

776

