SAMMET

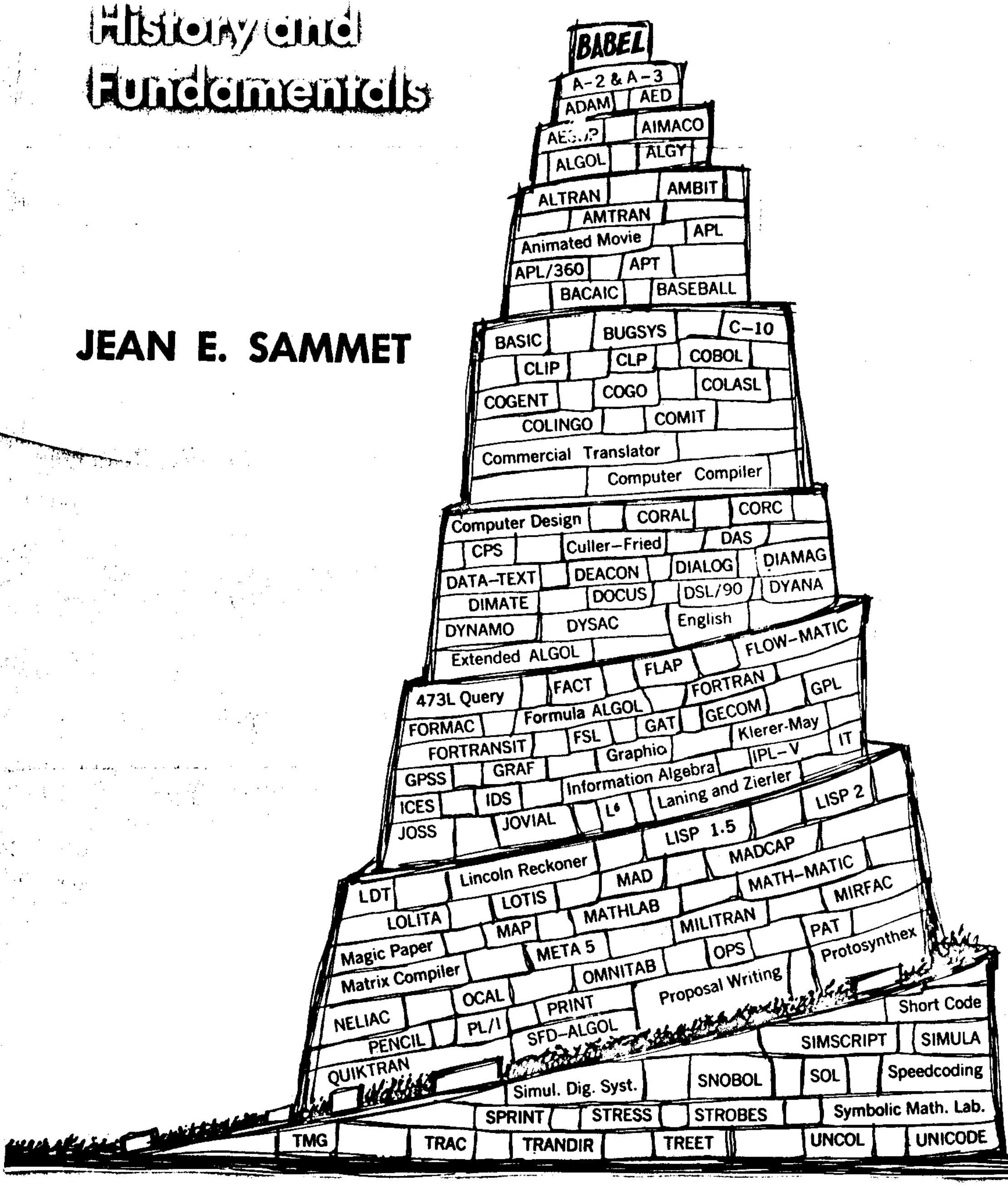
PROGRAMMING LANGUAGES:
History and Fundamentals

Prentice-Hall

PROGRAMMING LANGUAGES:

History and Fundamentals

JEAN E. SAMMET



PRENTICE-HALL SERIES IN AUTOMATIC COMPUTATION

SAMPLE PROGRAM—JOVIAL

Problem: Construct a subroutine with parameters A and B such that A and B are integers and $2 < A < B$. For every odd integer K with $A \leqslant K \leqslant B$, compute $f(K) = (3K + \sin(K))^{\frac{1}{2}}$ if K is a prime, and $f(K) = (4K + \cos(K))^{\frac{1}{2}}$ if K is not a prime. For each K, print K, the value of $f(K)$, and the word PRIME or NONPRIME as the case may be.

Assume there exists a subroutine or function PRIME (K) which determines whether or not K is a prime, and assume that library routines for square root, sine and cosine are available. This program also assumes the existence of three output routines. [Note: JOVIAL has ODD as a primitive.]

Program:

```
PROC      SPEC (A1, B1)$  ITEM A1 I U 47 $
                          ITEM B1 I U 47 $
BEGIN     WRITE (0) $
          IF NOT ODD (A1)$
          A1 = A1 + 1$
          FOR K = A1, 2, B1$
  BEGIN       WRITEN (15, K, 0)$
    IFEITH    PRIME (K) $
   BEGIN  WRITEN (30, SQR, (3*K + SIN (K)), 5)$
          WRITEH (45, 5H(PRIME))$
   END    ORIF 1 $
   BEGIN  WRITEN (30, SQR (4*K + COS (K)), 5) $
          WRITEH (45, 8H(NONPRIME))$
   END
  END     WRITE (1) $
END       WRITE (4) $
END
```

The language serves simultaneously as a reference, publication, and hardware language. JOVIAL was designed for the professional programmer and definitely to be used in a batch environment. However, a much later and much simpler version called JTS (see Sandin and Foote [SN65]) was installed under SDC's time-sharing system, and an interpretive extended subset version called TINT was specifically designed and implemented for on-line use. (See Kennedy [KE 65].)

JOVIAL has had the misfortune to suffer throughout its history from all the problems that could possibly arise from an attempt to have wide usage, maintain compiler independence, avoid dialects, and control subsetting and extensions. The proliferation of documents and systems on differing machines did not help the situation, although there were continuous attempts in SDC to control this problem. The earliest description seems to be the one by Schwartz, Petersen, and Olson [SC60]. The reader interested in pursuing which versions existed on which machines should see the papers by Shaw [SH63b] and Steel [ST66], but even these are not complete. (See also Figure

VIII-1.) There have been sev
the latest one by Perstein [PE
more positive side, in August,
were made in SDC with the r
any new JOVIAL compiler mu
manual. If the new comp
(JOVIAL 3), it must implemen
tion given in Perstein [PE66a].
are not included in the specifi
with the specifications of JOVI
dependent, e.g., precision of
definitions have been given in
little problem of incompatibili

Until 1967 there was no s
viewpoint of American stand
obviously there has been tre
itself. As a result of interest by
that a USASI standard might

The original CLIP langu
Book, and the former super
system.[4] Since then, numerous
further development of JOVIA
directly involved with the imp
groups to control the mainten

The basic objective of th
by professional programmers
cessing problems. In the vari
notations for defining syntax
arbitrary notation that in my
whatsoever, the latter appeari

One complaint which n
activity is a shortage of docum
ments of JOVIAL weighs almo
Since that was written in 1962,
decreased in quantity. Natur
papers of interest to limited gr
of it is of widespread intere
Kennedy [KE62]) to detailed
to general description and tut
Other references are listed at

[4] Shaw [SH00].

[5] Shaw [SH63b], p. 90.

750

nent is the creation and clear usage
e language (*SL*) and intermediate
has been entirely about the source
embles ALGOL 60. On the other
to LISP 1.5. *IL* is designed to retain
am to have the same structure as
he user and system programs. Thus
of having programs look like data,
uage level. There is a macro expan-
t the source language level.
operations which are useful com-

lf onto a new machine, and in fact
. In any case, since earlier versions
, this facility can certainly be con-
age level and, presumably, also at

language on the compiler is the
e, because a person who merely
d no background whatsoever of
arily tend to develop an internal
ing that has been done to improve
single words rather than to pack
bage collection in LISP, it was
on of this issue in Weizenbaum

al usage, no comment along those
if any—further work will be done
mediate change.

ES

a Unification ALGOL-COBOL,
e Question of One or Several Lan-
ata Processing, Gordon and Breach,

bining ALGOL and COBOL", *Proc.*
.

s of 8/67)", *Appendix 1, Minutes of*
7).

[AF67] *Standard Computer Programming Language for Air Force Command and Control System* (CED 2400), Air Force Manual AFM 100-24 (June, 1967).

[CE67] Clark, E. R., "On the Automatic Simplification of Source-Language Programs", *Comm. ACM*, Vol. 10, No. 3 (Mar., 1967), pp. 160–65.

[CO61] Coffman, E. G., Jr., *A Brief Description and Comparison of ALGOL and JOVIAL*, System Development Corp., FN-5618, Santa Monica, Calif. (June, 1961).

[KE62] Kennedy, P. R., *A Simplified Approach to JOVIAL* (A Training Document), System Development Corp., TM-780/000/00, Santa Monica, Calif. (Sept., 1962).

[KE65] Kennedy, P. R., *The TINT Users' Guide*, System Development Corp., TM-1933/000/02, Santa Monica, Calif. (Mar., 1965).

[KK65] Klein, S., "Automatic Paraphrasing in Essay Format", *Mechanical Translation*, Vol. 8, Nos. 3 and 4 (June, Oct., 1965), pp. 68–83.

[MD64] Marsh, D. G., "JOVIAL in Class", *Annual Review in Automatic Programming*, Vol. 4 (R. Goodman, ed.). Macmillan, New York, 1964, pp. 167–81.

[MI62] *FAST—FORTRAN Automatic Symbol Translator* (reference manual), MITRE Corp., SR-24, Bedford, Mass. (Jan., 1962).

[PE66] Perstein, M. H., *Grammar and Lexicon for Basic JOVIAL*, System Development Corp., TM-555/005/00, Santa Monica, Calif. (May, 1966).

[PE66a] Perstein, M. H., *The JOVIAL (J3) Grammar and Lexicon*, System Development Corp., TM-555/002/04, Santa Monica, Calif. (May, 1966).

[SC60] Schwartz, J. I., Petersen, K. E., and Olson, W. J., *JOVIAL and its Interpreter, A Higher Level Programming Language and an Interpretive Technique for Checkout*, System Development Corp., SP-165, Santa Monica, Calif. (Apr., 1960).

[SH00] Shaw, C. J., *Programming Languages and JOVIAL*, System Development Corp., BR-3/11-60, Santa Monica, Calif.

[SH61] Shaw, C. J., "System Development Corporation's Procedure-Oriented JOVIAL", *Datamation*, Vol. 7, No. 6 (June, 1961), pp. 28–32.

[SH61a] Shaw, C. J., *The JOVIAL Manual*, pt. 3, *The JOVIAL Primer*, System Development Corp., TM-555/003/00, Santa Monica, Calif. (Dec., 1961).

[SH63] Shaw, C. J., "A Specification of JOVIAL", *Comm. ACM*, Vol. 6, No. 12 (Dec., 1963), pp. 721–36.

[SH63a] Shaw, C. J., "JOVIAL—A Programming Language for Real-time Command Systems", *Annual Review in Automatic Programming*, Vol. 3 (R. Goodman, ed.). Pergamon Press, New York, 1963, pp. 53–119.

[SH63b] Shaw, C. J., "JOVIAL and Its Documentation", *Comm. ACM*, Vol. 6, No. 3 (Mar., 1963), pp. 89–91.

[SH64] Shaw, C. J., *A Comparative Evaluation of JOVIAL and FORTRAN IV*, System Development Corp., N-21169, Santa Monica, Calif. (Jan., 1964).

[SN65] Sandin, N. A. and Foote, E. B., *JTS User's Manual*, System Development Corp., TM-1577/000/01, Santa Monica, Calif. (Apr., 1965).

[ST66] Steel, T. B., Jr., *Some Observations on the Relationship Between JOVIAL and PL/I*, System Development Corp., TM-2930/000/01, Santa Monica, Calif. (May, 1966).