

# IEEE Standard for Programmed Inquiry, Learning, or Teaching (PILOT)

Sponsor

**Microprocessor and Microcomputer Standards Subcommittee  
of the  
IEEE Computer Society**

Approved June 27, 1991

**IEEE Standards Board**

**Abstract:** Language and coding standards for the PILOT programming language are described. PILOT (Programmed Inquiry, Learning, Or Teaching) is a language for the development of conversational interactive sequences often used for the computer-based instruction. The intent is to improve code portability and to minimize the development of dialects of this language, thereby improving the sharing of courseware among teaching institutions. PILOT syntax, core statements, and the core language are defined. Variables and language extensions are covered.

**Keywords:** computer-assisted instruction; computer-based instruction; PILOT programming language; programmed inquiry, learning, or teaching; self-evaluation testing

---

The Institute of Electrical and Electronics Engineers, Inc.  
345 East 47th Street, New York, New York 10017-2394, USA

Copyright © 1991 by the Institute of Electrical and Electronics Engineers, Inc.  
All rights reserved. Published 1991. Printed in the United States of America

ISBN 1-55937-151-X

*No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise without the prior written permission of the publisher.*

**IEEE Standards** documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other way to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five year for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change text, together with appropriate supporting comments.

**Interpretations:** Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation request except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board  
455 Hoes Lane  
P.O. Box 133  
Piscataway, NJ 08855-1331  
USA

IEEE Standards documents are adopted by the Institute of Electrical and Electronics Engineers without regard to whether their adoption may involve patents on articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.

## Foreword

[This foreword is not a part of IEEE Std 1154-1991, IEEE Standard for Programmed Inquiry, Learning, or Teaching (PILOT).]

## Historical Perspective

The PILOT language developed as a result of experience with an interactive computer system called COMPUTEST, developed at the University of California in San Francisco in 1962 in collaboration with the Dixie Elementary School District in Marin County. This system ran on a small IBM 1620 computer and was able to carry out interactive programs with one user at a time by means of its console typewriter. In early versions, the COMPUTEST system operated with program information (the program written in COMPUTEST) stored in a deck of punched cards and read into the computer a few cards at a time when the program was executed. Later, extended versions of the system provided greatly increased capabilities by using disk-storage units that allowed random access to the program material.

When an IBM 360 model 50 computer with time-sharing capability became available at the UCSF campus, a new compute-assisted instruction system called PILOT was developed. The language features of the PILOT system were patterned after those of COMPUTEST and stressed easy entry of instructional material by the program author. It was felt that a teacher should not have to become a computer expert in order to develop a computer-assisted instructional program. The PILOT system has been used for traditional frame-oriented instructional program as well as for self-evaluation testing, practice with simulated clinical interviewing, and a number of applications that attempted to respond appropriately to natural language input from students.

Not long after the original version of the PILOT was written for the IBM 360 computer, several versions of the language were programmed to run on other computers, some of them using more general-purpose languages, such as BASIC, SNOBOL, PL/1, or an assembly language specific to the computer. Some were called PILOT, and some had derivative names such as PYLON and NYLON.

Partly because of the requirements of different computer systems and partly because of individual preferences, each version of the language was somewhat different from the others. In early 1973, representatives of six of the major versions of the PILOT language met together and agreed on a set of core language specifications to be common among all the systems. This language represented the experience of many workers in computer-assisted instruction at that time. Recognizing that differences would undoubtedly arise, there was also agreement on a standard means of describing functions outside the core language. Because many of those involved had applications in elementary and secondary schools settings, the resulting standard, called PILOT 73, was intended to be language that could be easy for the program author to learn.

A major dialect of PILOT that departed from some of the PILOT 73 guidelines was developed at Western Washington State College in Bellingham, Washington. It is called COMMON PILOT, and has extensions and additional functions to handle the more complex programming needed in college-level instruction, particularly in mathematics and science. This version of PILOT has formed the basis of PILOT for the Apple microcomputer, for Radio Shack microcomputers, for C-Pilot on UNIX systems, and PC/PILOT for the IBM Personal Computer.

PILOT 73 was developed at UCSF for the Datapoint computer and became the basis of versions for an early microcomputer, The Processor Technology Sol. Before this company went out of business, it published a cassette version of PILOT, and a version for the company's Helios disk system was in preparation. These were distributed by PROTEUS, The Processor Technology User's Group.

A version that matched most PILOT 73 requirements was called Nevada PILOT and ran on the CP/M operating system. A related version called Utah PILOT was developed for MS-DOS. The Atari company developed PILOT for its microcomputer in a version that is easy for elementary students to use. It contains graphic extensions providing the kind of "turtle" graphics found in LOGO. Although PILOT itself does not use line numbers, the Atari version requires them for program entry and editing, using the same mechanisms as provided for BASIC.

## Development of the Standard

The Technical Committee on Microcomputers and Microprocessors of the IEEE Computer Society approved a project authorization request (PAR) for PILOT in July 1987. A working group was assembled from known implementers of PILOT versions, those involved in the support of user groups, and others who expressed an interest. Notices of the working group activity were placed in appropriate newsletters. A first and general draft of core standards was circulated in March 1988. It was a one-page description of the purpose of the PILOT language, the purpose of the standard, and a list of core statements. This was extended in a second draft of June 1988, which included a formal definition of core PILOT in Backus Normal Form (BNF).

An alternative BNF description was provided by the designer of PILOT for the Amiga computer, a version that reflected considerable compatibility with PC/PILOT. A document was prepared to compare the two BNF descriptions and indicate areas needing resolution. In September 1989, the working group held a meeting in conjunction with IEEE Standards Week in San Francisco. Problems were listed and approaches to solving them were discussed. In November 1989, the list of problems and proposals for solution were circulated to all members with a request that any disagreements be brought to light.

At the time this standard was completed, the Working Group on PILOT (1154) had the following membership:

**John A. Starkweather**, *Chair*  
**Richard H. Karpinski**, *Secretary*

Glen Chapman  
Jim Flournoy  
William H. Ford, Jr.  
George Gerhold  
Ted M. Khan

Larry Kheriaty  
Terry LaGrone  
Brian L. Meek  
Thomas G. Nielsen  
Richard J. O'Connor  
W. Kirk Richardson

Roy A. Standing  
Thos Sumner  
Dave Taylor  
Charles S. Tidball  
Carl Zinn

The following persons were on the balloting committee that approved this document for submission to the IEEE Standard Board:

Charles Brill  
Winsor A. Brown  
Myron Calhoun  
Wayne Fischer  
Gordon Force  
Patrick Gonia  
Zoltan R. Hunor  
Jack R. Johnson

Richard H. Karpinski  
Thomas M. Kurihara  
James D. Mooney  
Cuong Nguyen  
Donald Pavlovich  
Shlomo Pri-tal  
Steven Ray  
Hans Roosli

Paul Rosenberg  
John Starkweather  
Manu Thapar  
Robert Tripi  
Carl Warren  
Fritz Whittington  
Andrew Wilson  
Oren Yuen

When the IEEE Standards Board approved this standard on June 27, 1991, it had the following membership:

**Marco W. Migliaro**, *Chair*  
**Donald C. Loughry**, *Vice Chair*  
**Andrew G. Salem**, *Secretary*

Dennis Bodson  
Paul L. Borrill  
Clyde Camp  
James M. Daly  
Donald C. Fleckenstein  
Jay Forster\*  
David F. Franklin  
Ingrid Fromm

Thomas L. Hannan  
Donald N. Heirman  
Kenneth D. Hendrix  
John W. Horch  
Ben C. Johnson  
Ivor N. Knight  
Joseph Koepfinger\*  
Irving Kolodny  
Michael Lawler

John E. May, Jr.  
Lawrence V. McCall  
T. Don Michael\*  
Stig L. Nilsson  
John L. Rankine  
Ronald H. Reimer  
Gary S. Robinson  
Terrance R. Whittemore

\*Member Emeritus

| CLAUSE                                                     | PAGE |
|------------------------------------------------------------|------|
| 1. Introduction .....                                      | 1    |
| 1.1 Scope.....                                             | 1    |
| 1.2 Purpose.....                                           | 1    |
| 1.3 Applications of the PILOT Language.....                | 1    |
| 1.4 Legal Disclaimer .....                                 | 2    |
| 2. Conventions .....                                       | 2    |
| 2.1 General PILOT Syntax.....                              | 2    |
| 2.2 PILOT Core Statements.....                             | 2    |
| 2.3 Formal Definition of Core PILOT .....                  | 3    |
| 3. Variables .....                                         | 6    |
| 3.1 String Variables.....                                  | 6    |
| 3.2 Numeric Variables .....                                | 6    |
| 4. Language Extensions .....                               | 7    |
| 4.1 Key-letter Extensions .....                            | 7    |
| 4.2 Key-word Extensions .....                              | 7    |
| 4.3 Modifiers.....                                         | 7    |
| 4.4 Dimensioning Strings.....                              | 8    |
| 4.5 Further Definition of the C: (Compute) Statement ..... | 8    |
| 4.6 Coding of Multiline Text .....                         | 8    |
| 4.7 Intraline Comments.....                                | 8    |
| 4.8 System Variables.....                                  | 8    |
| 4.9 Extensions to J: (Jump) Using '@' .....                | 9    |
| 4.10 Further Extensions .....                              | 9    |