PROGRAMMING LANGUAGE ONE (PL/I)

By:

Ankit Jain, Roll no. 15
Amol Deodhar, Roll no. 25
History

PL/1 was developed as an IBM product in the mid 1960's, and was originally named NPL (New Programming Language). The name was changed to PL/1 to avoid confusion of NPL with the National Physical Laboratory in England. If the compiler had been developed outside of the United Kingdom, the name may have remained PL/1.

Until the time this new language was developed, all previous languages had focused on one particular area of application, such as science, artificial intelligence, or business. PL/1 was not designed to be used in the same way. It was the first large-scale attempt to design a language that could be used in a variety of application areas.

Significant Language Features

PL/1 had the following significant language features:

- PL/1 is completely free form and has no reserved keywords.
- It precisely defines its data types without regard for any hardware.
- PL/1 is a block-oriented language consisting of packages, begin blocks, and statements. This type of structure allows the programmer to produce highly-modular applications.
- PL/1 contains control stuctures. For example, SELECT...WHEN...OTHERWISE allow logical operations, and DO statements allow units to be executed unconditionally one time, forever, or while a condition is true or until a condition becomes true.
- PL/1 supports arrays, structures, unions, arrays of structures or unions, structures or unions of arrays, and combinations thereof.
- PL/1 provides four different storage classes: AUTOMATIC, STATIC, CONTROLLED, and BASED. Application objects' data type, representation, nature of use, etc. normally decides the type of storage class used for each.
A simple program in PL/I

**EXAMPLE 1:** This program demonstrates the text output function of the PL/I programming language by displaying the message "Hello world!"

**Source Code**

HELLO:   PROCEDURE OPTIONS (MAIN);

    /* A PROGRAM TO OUTPUT HELLO WORLD */
    FLAG = 0;

LOOP:     DO WHILE (FLAG = 0);
    PUT SKIP DATA('HELLO WORLD!');
    END LOOP;

END HELLO;

**EXAMPLE 2:** Stripc :strips preceeding characters (e.g. blanks).

StripC: proc( inStr, inChar ) returns( var char(255) ) recursive;

dcl inStr char(*),
    inChar char;

dcl substr builtin;

    if substr( inStr,1,1 ) = inChar
      then return( StripC( substr( inStr,2 ), inChar ));
    else return( inStr );

end StripC;

PL/I has a very large vocabulary of built-in functions. In fact, there is probably no one compiler that has the full standard of keywords available. PL/I compilers are normally subsets of the language that specialize in various fields.
Partial list of features:

- Preprocessing
- Full support for pointers
- Case-insensitive keywords
- Variable-length arrays
- Call by reference is the default
- Supports complex structure declarations with unions
- Built-in support for a slew of data types, including two types of strings
- Four classes of storage: Automatic, Static, Controlled (dynamic) and Based (anonymous dynamic)
- Automatic garbage collection
- Built-in coprocessing facility

Business users mainly used COBOL, while scientific users used Fortran. The goal of PL/I was to develop a single language usable for both business and scientific purposes. Another goal was to add structured programming constructs derived from ALGOL, which neither COBOL nor Fortran supported (at the time). PL/I was designed by a committee drawn from IBM programmers and users drawn from across the United States, working over several months.

GENEALOGY of PL/I

References:

http://www.engin.umd.umich.edu/CIS/course.des/cis400/pl1/pl1.html#sigfeat

http://hjem.get2net.dk/micrask/pli/pli.html

http://home.nycap.rr.com/pflaspl.htm

http://www.users.bigpond.com/robin_v/pliwhat.htm