TIF : A Program Interpreter in Filipino

A Special Problem

Presented to

the Faculty of the Department of Mathematical Sciences and Computer Studies

De La Salle University - Dasmariñas

Dasmariñas, Cavite

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

Rodel M. Talampas

March 1997

ABSTRACT

NAME OF INSTITUTION : De La Salle University - Dasmariñas

ADDRESS : Area B, Dasmariñas Bagong Bayan, Dasmariñas, Cavite

TITLE : TIF : A Program Interpreter in Filipino

AUTHOR : Rodel Manalo Talampas

FUNDING SOURCE : Personal COST : 3,000.00

DATE STARTED : November 30, 1996 DATE COMPLETED : January 30, 1997

OBJECTIVES OF THE STUDY

A. GENERAL : To create a General Purpose Filipino Program Interpreter

B. SPECIFIC : To adopt this Interpreter as a Beginner's Language

SCOPE AND COVERAGE :

TIF is a programming language interpreter that uses the seven basic fimdamental elements of programming as its foundation. A programmer can create programs that uses basic input/output operations, the assignment statement, the conditional statement, the subroutines, the iterative statements, the arithmetic operators, and can create procedures of his own.

METHODOLOGY :

The author used the principles embodied in the BNF, Syntax Graphs, and Parsing in creating his own Program Interpreter. He used the prototyping design method in developing the software.

OUTPUT OF THE STUDY :

The software developed was a Tagalog Interpreter. It was created using the traditional techniques and methods of programming language design. A complete set of the requirements for an interpreter were addressed in the software. It was created for Filipinos to participate in the growing field of computer programming.

CONCLUSIONS :

A Program Interpreter was developed using the traditional techniques of programming language design namely : BNF, Syntax Graphs and Parsing. It recognized the seven fundamental elements of programming needed for the efficiency of the program being created.

The author concluded that creating a language interpreter is not hard as it is. All you need is patience, sacrifice, self-discipline, and hardwork.

RECOMMENDATIONS :

The author recommends that whoever is willing to continue his work must consult him and he is willing to help. Further work includes: a text editor, unlimited number of variables and can handle different kinds of data types.