CHAPTER 1

INTRODUCTION

1.1 BACKGROUND:

Programming, like any other skill and creative work, because of the creative work, it involves two of processes "DESIGN" and "ART". The term "DESIGN" means the application of established computer techniques to produce equipment that will conform to a given specification, the term "ART"means the techniques to arrange the equipment to be easy to read, easy to understand and easy to modify;.

After the programmer received the program specification which briefly indicate its purpose, then he must translate it into a logical process and the set of computer instructions for operation of the given step within a system. The documentation must therefore be developed as the program is being written so that it can be understood by others at the time of writing and later when it has to be modified or to be amended. The stage which this documentation should take place namely:

- Program Outline Flowchart (WHAT THE PROGRAM DOES)
- Program Detail Flowchart (HOW THE PROGRAM WORKS)

Up to now, we cannot specify the fixed standard set of rules in program writing. Each program has its own special difficulties and there are no formulae that can deal with them all. The programmer also has the individual programming style according with his skill and experience. It causes that no one can know others program, many conventional programming seems to be based on the concept; "Program as best you can and then test the dickens out of it", after the program already used as the production, the bug may come when the uncovered condition found. With this reason, the program documentation is very important tool to aid the program amendment, program updated and program modification. Without the program documentation will extremely expensive both in terms of time and money and it is even possible that the program may have to be re-written.

1.2 THE PROBLEM IN PROGRAM DOCUMENTATION:

The problem may be classified into 2 stages, the document providing stage and the document handling stage.

The document provided stage:

- The programmer finds himself at the end of the job with a demoralizing amount of clerical work to be done which appears the "Unproductive". The nature desires to move onto a new program may tempt the programmer to provide incompleted documentation which will clearly be of little value.

- Obviously, if several programs are required very urgently, the temptation to ignore the documentation and simply to concentrate on producing a program which work is very great.

- The programmer always jumps to program coding before program documentation. When the program already work, he may forget the procedure steps in the program and when he comes to do the documentation, the important points may be overlooked.

The document handling stage:

- Frequently, the program documentation was not updated or incorrectly updated after the production program was amended or modified. It causes the program documentation becomes to the unvaluable history document.

- Sometimes, the program documentation may be lost or may be damaged with the careless and neglectful handle.

1.3 OBJECTIVE OF THIS THESIS:

The objective of this thesis is to provide a tool in order to

- help the programmer or the analyst to create the program outline flowchart.
- generate the cobol source program, not only the ANSI cobol but also several kinds of cobol program into the program detail flowchart.

The tool is a program written in some basic statements of ANSI cobol language. Since the cobol language is widely used in computer data processing and is approved by the American National Standard Institute to be a standard language, therefore, this program can be operated on several general purpose computer.

The program also provides 4 additional requirements included in flowchart generating:

New verbs added:

This feature allows the user to submit new or specific cobol verbs and his general data processing verbs, its purpose is for the ability of the program to generate the flowchart of any specific cobol source program, ANSI cobol extended source program and the user's outline text.

Unaffected flow verbs deleted:

It is provided for deleting the unrelated logic flow verbs like the source program listing control, debugging, compiler directing verbs, out of the program flowcharting report.

New more meaning dataname:

To abandon the unclear or no meaning dataname in favor of more representative names. The name likes "A-NO" were abandoned for more meaningful tags like "ACCOUNT-NUMBER" and some unclear names like "KEY" were replaced by more specific name like "TRANSACTION-KEY". The replaced and replacing names are listed at the end of flowchart.

Document title:

The users can submit the title of the flowcharting report in order that making it well document. The title lists are:

Program Explanation	:	Specification, name and number, date created.
Program amendment and modification	:	Reason to change, changed number and date changed.
Responsibility name	:	Analyst, Programmer, Approved name.

1.4 THE SEQUENCE OF THE CONSTRUCTION METHODS:

1.4.1) The flowchart symbols and structure studying.

1.4.2) Cobol grammar and structure studying.

1.4.3) Algorithm concept and method studying.

1.4.4) Tool designing.

1.4.5) Program writing and coding.

1.4.6) Program compilation and testing.

1.4.7) Final documentation.

1.5 THE BENEFIT OF THE TOOL:

1.5.1) With the outline flowchart generating benefit:

- Reducing the documentation clerical work.

- Helping the conversion from program specification to program coding easy.
- Preventing the outline document from losing and damaging, it can be reprinted from computer any times.

1.5.2) With the cobol program flowchart generating benefit:

- Making the cobol source program easy to read since its diagrammatic.
- Making cobol source program intelligible to others.
- Using it as a helping tool in program development, amendment and modification stage.
- It's always up-to-date since it is generated from source program.