NIGERIA JOURNAL OF BANKING FINANCE AND MANAGEMENT , VOLUME 12, NUMBER 1, JANUARY 2011

COMPUTER PROGRAMMING

BY

DR. OGOLO JOHN IBINWANGI
DEPARTMENT OF BUSINESS ADMINISTRATION
NIGER DELTA UNIVERSITY
WILBERFORCE ISLAND
BAYELSA STATE OF NIGERIA

ABSTRACT

A computer is a machine that is capable of using programs, instructional and data to provide managerial decision making. It has a process whose input is data and its output is information. By data is meant a representation of facts, observations and occurrences.

A process has steps for transforming inputs to output. Computer Programming is a process for using a language to develop the programs. There are a myriad of programs like FORTRAN, COBOL, Basic, AppI, C, JAVAL, C^{++}, etc. Some of them are now obsolete and some have different versions like Fortran IV, Fortran V. etc.

INTRODUCTION

At its most basic level, computer programming simply means telling the computer what to do and the rapid sounding definition is not even a joke. There are no other truly fundamental aspects of computer programming; everything else we talk about will simply be the details of a particular, usually artificial, mechanism for telling a computer what to do. Sometimes these mechanisms are chosen because they have been found to be convenient for programmers (people) to use; other times they have been chosen because they are easy for the computer to understand. The first hard thing about programming is to learn, become comfortable with and accept these artificial mechanisms, whether they make sense to you or not (Starkey et al, 1984).

In fact, it doesn't matter of some (or even many) of the mechanisms used for programming a computer don't make sense. After all, it doesn't make sense that the cold water faucet has to be on the right side and the hit one has to be on the left; of course, that is just the convention we have settled on. On a similar note, many computer programming mechanisms are quite arbitrary and were chosen not because of any theoretical motivation but simply because we needed an unambiguous way to say something to a computer (Walraet, 1989).

Programming requires mainly a logical sense of reasoning; an ability to supply a set of instructions (the program) so that the computer can execute them. In the early days of computer programming, specialists used a language now called low level programming language to communicate with the computer. This language is very much like a set of instructions given in binary arithmetic. For instance, during first generation computer (1951 — 1959) programming was tedious since it was done in machine language. This was characterized by the use of vacuum tube technology or their electric circuitry (Ehiemeke et al, 2004).

During second generation (1959 — 1965), programming was done in symbolic (assembly language) rather than machine language. Assembly language involves the use of mnemonics in writing program. A translator called Assembler was put in place to translate program written in assembly language into machine language instruction code. This generation was characterized by the use of transistor. The use of high-level languages in writing programs came about shortly after the zenith of the second generation. This involves the use of English like terms in writing programs. From 1966 till date, several programming have been invented among which are FORTRAN, COBOL, BASIC, C, C^{++}, JAVAL etc (Oladipupo et al, 2006).

CONCEPT OF COMPUTER PROGRAMMING

The concept of computer programming is a cross fertilization of two terms" Computer and Programming. A computer is an electronic device that operates under the instruction & a stored program, accepts inputs (data), processes it according specified rules, produces output (information) and stores the result for future use (Ehimeke et al, 2004).

A computer system is made up of two main parts: hardware and Software. Hardware is the physical component of computer while software is a series of instructions that tell hardware how to perform tasks (Donovan, 2972). A program includes all the instructions and data, which must be provided to carry out a given, assignment. It is a set of instructions written in a codified form, which enables the computer to carry out its operational activities. A program could also be referred to as software. The process of preparing such program is called programming (Laplante, 1996). Computer programming is the process of written instructions to guide computer operations. It involves a step-by-step list of instructions written for particular computer architecture in a particular computer programming language (Mandell, 1986).

COMPUTER PROGRAMMING PROCESS

Computer program is a step-by-step list of instructions written for particular computer architecture in a particular computer programming language. A layman equivalent example will be writing a step-by-step list of instructions in English, instructing a student to register for courses for the academic session (the human being the specific architecture). More often than not cc.npu programs are compiled or assembled into non-human format. In other words, computer program can also be said to a process of taking the steps in a procedure and putting them in the coded language of the computer (Venit et at, 1998 and Mandell, 1986).

In the process producing the necessary instructions making up a program, the following stages can be recognized as can be seen in the diagram below:

1) **Understanding the Problem:** This is the first stage in trying to write a program. At this stage, the problem is identified, analyzed and must be properly understood.

2) **Planning the Method of Solution:** After understanding the problem to be solved, plans are made on what method to go about solving the problem.

3) **Writing the Instruction in a Programming Language:** At this stage the programmer will select the program language that best suits his or her requirements The instruction can be written in high level language, low level language, low level machine code (language).

4) **Transcribing the Instruction into a Machine Sensible form:** This is also known as the compilation stage. A program which has been written in a high level language must either be:

(i) Compiled into a machine code form, for input into the computer.

(ii) Input into its existing form to the computer, which will translate the instruction into machine code form using an interpreter.

(5) **Testing the Program:** At this stage the program that has been written will have to be tested. The programmer must test the object program with the test data. A diagnostic routine or debugging routine provides for routine program testing and error correction during program development. When the program does not run well, the error must be located. Diagnostic routine will find out what the program was doing at the time it failed, and what the contents of store were. Test packs are used to test the effective working of new computer programs before they are sued operationally.

(6) **Documenting all the work involved in producing the Program:** Starting from the point of understanding the program to be solved, to the point where the program becomes effective is documented.

(7) **Maintenance of the Program:** At this stage the workable program must maintained. This is important because the computer user may decide that I program ought to be capable of doing additional processing routine or six process some data in a different way

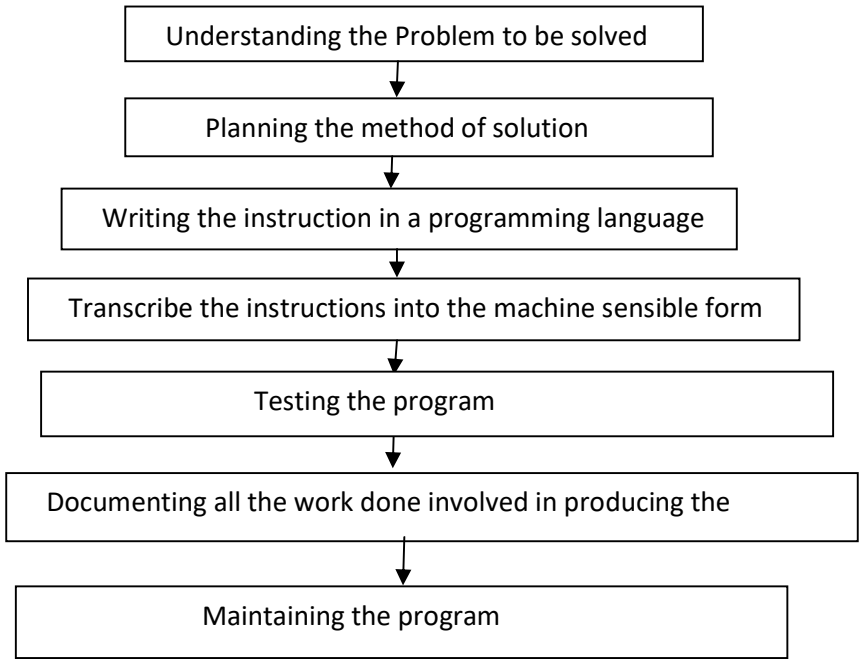


Figure 1: Stages in computer programming

Source: Oladipupo, A.O. et al 2006, page 108.

TOOLS IN COMPUTER PROGRAMMING

The following are the tools, which the computer programmers adopt in writing computer

1) Algorithm

2) Flowcharts

3) Decision table

1) **Algorithm**: It is a listing of the sequence of steps in their proper order of occurrence of the actions, which are necessary for the successful execution of a given task. Essentially, an algorithm is a finite series of logical steps required to solve a particular problem.

An example of the use of computer algorithm: For instance, an algorithm for a simple payroll program in which the variables: employee name, monthly salary, allowance and income tax are supplied as inputs. There is a possible algorithm for the above task.

1) Let initial value of total salary be zero.

2) Let initial value for counter be 1.

3) Read an employee record containing employee name, monthly salary, allowances and income tax.

4) Calculate net salary as: Net salary = Monthly salary + allowances — income tax

5) Write a payroll record for this employee printing employee name and net salary.

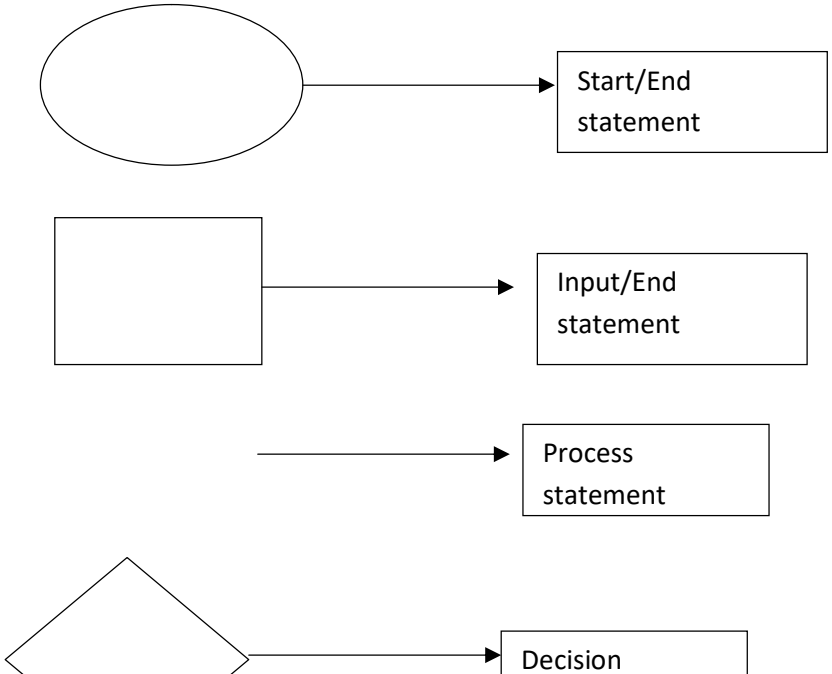
6) Add net salary to the current value of total salary.

7) Add 1 to current value of counter.

8) If the new value of counter is less than or equal to 2500, go back to s

9) Print the counter value of total salary.

2) **Flowcharts:** A flowchart is a pictorial or graphical tool or model that represents the events in a process. Some of the symbols of flowchart are given below:



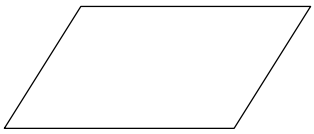


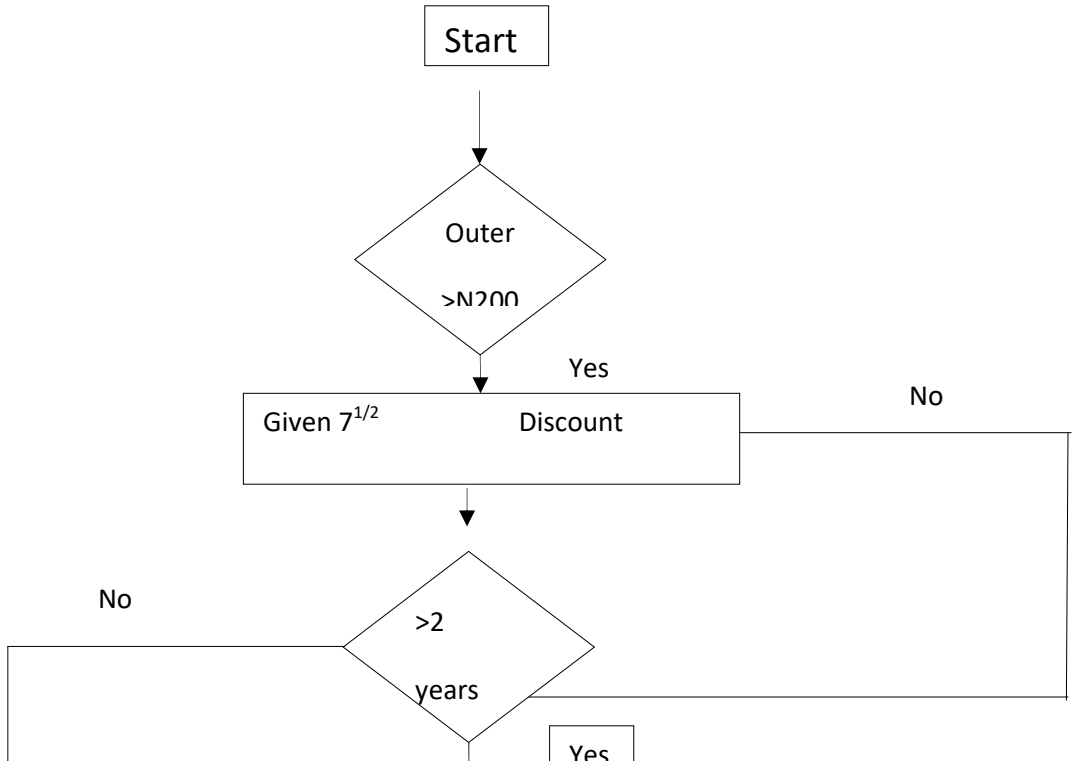
Figure 2: Some flowchart systems

Source: Oladipupo A.O. et al. (2006) page 110.

ILLUSTRATION OF FLOWCHART

When invoicing customers, the invoice clerk has worked out the each order. Any order over N20,000 attracts a bulk discount of 71/2%. A customer within the track is allowed 5%. There is also a special 5% allowed for any customer who has been ordering regularly for over 2 years.

Required: Construct a flowchart



igure 3:

Oladipupo A.O. et al (2006), page 111

3) **Decision Table:** Decision table is a tabular method of communicating the action which is appropriate to each of a set of a given condition. A decision table thus conveniently summaries complicated decision procedures. It is divided into four sections:

(a) The upper two parts are concerned with the conditions.

(b) The lower two parts are concerned with the actions.

Condition stub	Conditions
Action stub	Conditions

Figure 4.

Source: Oladipupo AD. et al (2006), page 112

FEATURES OF A GOOD COMPUTER PROGRAM

The following are some characteristics of a good computer program.

1) **Reliability:** This characteristic implies that the program can be depended upon always to perform efficiently.

2) **Maintainability**: That is the program should be easily modified and updated when the need arises.

3) **Profitability:** That is the program should will be transferable to a different computer with little or no modification.

4) **Readability:** This is to say the program should be easy for a programme to read and understand.

5) **Performance**: The program should be capable of causing the task to be performed quickly and efficiently.

6) **Storage Saving:** The program should not be allowed to be unnecessary along (Oladipupo et al (2006); Szymanski et al, 1988).

COMPUTER PROGRAMMING LANGUAGE

Programming languages are various by which the programs are written. Each program is written in a particular language, but there are quite a large number of different languages, just as spoken languages include English, French, German, Russia and Cantonese and so on. When a program is written, it is written in a programming language. This is neither the normal written language of humans, nor it is usually the machine code language of computers (Venit et al, 1988).

TYPES OF PROGRAMMING LANGUAGES

Programming languages can be grouped into 3 broad categories viz:

1. Low level languages

2. High level languages

3. Fourth generation languages

1. **Low Level Language:** These languages consist of machine assembly language.

a. **Machine Language:** It forms the earlier program developed by the computer programmer. It is generation. It uses the binary codes in digits of zero and in machine codes (languages) was usually very complex, prone, due to the fact that every program must be in the codes (language) before any task can be executed.

b. **Assembly Language:** Assembly language was a from machine code (language). This is a machine specific language, but the task of learning and writing the language is made easier than with the machine code (language), because it is written in symbolic form. instead of using machine code (language) operation numbers, the programmer is able to use learned and understood operation mnemonics (for example, ADD, SUB, MULT etc) and symbolic data locations (names allocated by the programmer and used to refer to particular data areas in the program) replace numeric data locations.

2. **High Level Languages:** These are called third generation languages. These are machine independent language which means they are compatible with any type of machine. This generation of computer languages uses extensive vocabulary of words and symbols are used to instruct a computer to carry out the necessary procedures, regardless of the type of machine being sued (although the program that translates a high level language into a machine code form known as a compiler or interpreter is different for each make of computer.

Examples of third level languages are:

a. FORTRAN - Formular Translator

b. COBOL - Common Business Oriented Language

c. BASIC - . Beginners All Purpose Symbolic Instruction Codes

d. ALGOL - Algorithmic Language

e. ADA - (World first Computer Programmer Augusta Ada Byron).

3. **Fourth Generation Language:** This generation of computer language has witnessed a market improvement on the existing computer languages. This generation of language is long, procedural and more conversional. The non- procedural aspect denotes that the user can instruct the computer on what he needs to achieve and the computer will generate the instructions to achieve the result (Brookshear, 1994; Oladipupo et al, 2003).

DOCUMENTATION DEBUGGING AND TESTING OF COMPUTER PROGRAMS

Program Documentation

A computer program without documentation would be like a teacher without records of

•students test scores or files of student's work. When students ask why they received certain grades, such as a teacher has no written documentation or student papers to back up the grades. When users or programmers wish to determine what program does and how, the program without documentation offers no written information (Cobumetal, 2000).

Documentation is the portion of a program that explains the program. It describes what the program should be, what data is needed, how data is identified in the program statements and how the output is formatted. In top down design, documentation describes the modules as well, explaining their individual functions and their relationship to the whole program. Documentation also includes the charts

produced during the program design, the tests and data used to check the program, and any changes made to the program. Other important documentation provides instructions to program users and lists hardware requirements for running the program (such as disk drives or printers). A listing of the actual computer program should also be included (Elsenbach, 1987).

Where does documentation appear? Some documentation appears in the actual program statements to identify the objectives of the major solution and the individual Modules. Other documentation may appear on the monitor when the program is run to help users understand what they should do with the program. Still other documentation appears in run books, which are used by the computer operator to run the program, and users' manuals, which should contain all documentation designed to aid users not familiar with a program. Users' manuals, for instance, explain how to establish contact with the computer and enter the commands to use software (Coffman Jr., 1992).

Proper documentation is a reference guide for programmers and analysts who must modify or update existing programs and procedures. Without it, a programmer may spend days or weeks trying to ascertain what a program does and how it does it. Further in many cases, programs are designed to operate under a fixed set of conditions and constraints. When organizations change and grow, program modification must keep pace with their changing needs. Documentation helps managers evaluate the effectiveness of the programs and determine where changes are desirable. Documentation is also essential to those who must perform the manual functions of entering data required by the system and running the program. The process of documentation is an ongoing one. It begins with the initial request for a program. It continues throughout the problem-solving steps and into the process of program maintenance (Lucas, Jr., 1986).

Debugging

Debugging is the process of locating, isolating and resolving errors within a program.

PROGRAM TESTING

Documentation is not the only process that should occur at each problem solve step. Each step should be tested before proceeding to the next step. During the early years of programming, testing often occurred after the entire program was completed. Corrections were made in a pitch-it-up manner, and major errors were often discovered after programs were implemented. Although errors still over-even after software is released for sale — software developers test their programs more thoroughly today than they did in the fledging years of the industry (Mandell, 1986).

Errors can be major (perhaps the program is incorrectly defined) or minor (a typing error causes an error message to appear in the monitor). Since difference kinds of errors occur at the various stages of program development, different types of testing are described or these errors (Coffman Jr., 1992).

The programmer must test the program by using adequate data to determine and ensure that the required outcome is achieved. The programmer uses senses of data to test the program in order to ascertain if the correct or desired results would be produced. If the program is seen not to be producing

accurate results, amendments are carried out and further testing conducted until the accurate or correct results are obtained (Oladipupo et al, 2006).

CONCLUSION

There is no gainsaying about the fact that computer programming is the basic building block around which any workable computer system is built. Hardware and software make up the basic components of a computer. Software is a suite of programs, which enable the computer to carry out its operations independent of any user-interface. Hence, without software (i.e. program), the computer will just be like a cabinet, which can be used for nothing else other than office decoration above assertion remains a truism because computer program encompasses various instructions and data, which must be provided to carry out a given assignment (Unyimadu, 2005). Strictly speaking, computer programming constitutes the process by which the codify instructions is a language understandable by the computers. The flow of a program sequence is often times represented by mans' of flow chart or algorithm. Several programming languages have been developed overtime vis-à-vis the various generations of computer. The low level languages (machine and assembly) had been successfully replaced by the invention of high level languages such as BASIC, FORTRAN, JAVA etc. (McGraucen, 1972).

Programming, as it were, is an Herculean tasks which involves critical mental thinking. Hence, programmers are expected to possess some special skills before they could develop a good software for problem solving. Program development, therefore could be a problem specific or user specific as the case may be. Quite a number of programs have been developed to solve variety of problems in business organizations (Szymanski et al, 1988).

In fact, the application of computer programs has completely . revolutionized our business/industrial sector. In most organization nowadays, electronic data processing has replaced the conventional manual data processing. A typical example is the Uniben Portal WAEUP (West African Electronic University Project), which has taken over the administration of Uniben. This was made possible under the auspices of a specific software designed for that purpose (Ehiemeke et al, 2004).

REFERENCES

Ait, F.L. (1958). Electronic Digital Computers, New York: Academic Press.

Bob, W. (1989). Programming the Impossible Challenge, Elsevier Science Publisher B.V.

Chiemke, S. and Egbokhare, F. (1998). Data Processing and Management Information System, Benin City: Ardiu Publishers, p. 32.

Coffman, Jr. E.G. (1992). Computing, North Holland.

Davis. W.S.C. (1991). Computing Fundamental Concepts, New York: Addison — Wesley Publishing Company pp. 195.

Edward, J.C. (2000). Programming with Visual Basic, Brooks/Cole Publishing Company.

Fajuyigbe. 0. (2002). Management Information System. Lagos: YBT Publishers.

Glenn, JR. (1994). Computer Science, Addison Wesley Publishing Company.

John, D.C. (1972). Systems Programming, McGraw-Hill Book Company.

Kro•eber. D.W. and Watson, H.J. (1987). Computer Based Information Systems — A Management Approach, New York: Macmillan Publisher Company, pp. 600.

layre. C. (1999). Data Processing — Practical Guide, Benin City: G.C. Consultancy Limited. Pp 194.

Lucas, H.C. (Jr.) (1986). Introduction to Computer and Information System, New York: Macmillan.

McGraucen, D.D.A. (1972). Guide to FORTRAN IV Programming. New York: John Wiley and Sons Incorporated.

McLeond. R. (1998). Introduction to Information Systems — A Problem-solving Approaching, Chicago: Science Research Associates, Incorporated.

Obasogie. D.C. (1998). Fundamental of Digital and Analogue Computer Design, Warn: Merit Publishers.

Oladipupo and llaboya (2006). Information Processing System (Theory and Practice), Benin City: Mindex Publishing.

Philip, L. (1996). Great Papers in Computer Science, West Publishing Company.

Radlow, R. (1986). Computers and the Information Society, New York: McGraw-Hill Book Company, pp. 456.

Starkey, J.D. (1984). Fundamental Programming, West Publishing Company.

Stella E.F and Egbokhare, G.E. (2004). Data processing and Management Information System. Ardix Publishers.

Steven. L.M. (1986). 2nd Edition: Computers and Data Processing Today, West Publishing Company.

Stewart, V. (1998). Programming in true Basic, PWS Publishing.

Szymanski, R.A (1988). Introduction to Computers and Information System, Menu Publishing Company.

Unyimadu, S.D. (2004). Introduction to Research Me;ods, Benin City: Harmony Publishers.