



INTRODUCTION TO PROGRAMMING

What is a software program? A software program is a series of instructions for the computer to obey.

STAGES IN PROGRAMMING

1. **Understanding the problem:** This is needed to know the correct program to solve the problem.
2. **Planning the solution:** This includes finding ways to transform the input into the desired output.
3. **Preparing the Algorithm:** The two methods of developing programs are; **Flow chart and Pseudocode** which will be explored further down
4. **Coding:** This translates the logic specified by flowchart and pseudocode into a statement in a specific language.
5. **Program Testing:** Once a program is created, it is tested for syntax and error. This is usually done by a program in the execution environment.
6. **Documentation:** This is a copy of the work performed on the program.

ALGORITHM

Algorithm is a sequential steps involved in solving problem.

Characteristic of Algorithm

1. It must be precise
2. It must have a termination
3. It must be effective
4. Should have starting Instruction to accept input





FLOWCHART

Is a diagrammatic representation of algorithm.

Importance of Flowchart

1. It serves as a guide for programmers
2. It is very good for illustration
3. It is a very good tool for modification during program update

Flowchart Symbol

1. **Terminal Symbol:** The oval shape symbol is used to express the beginning and end of an algorithm



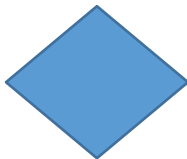
2. **Input/Output Symbol:** The parallelogram symbol is used to express processing operation.



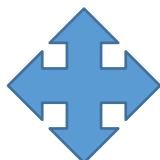
3. **Processing Symbol:** The rectangular shape is used to express processing operations



4. **Decision Symbol:** For making decision. It usually contains True Or False



5. **Arrow symbol:** To control the flow of the program



6. **Connector symbol:** This is used to connect more than one sheet of programs together for long codes that contains several sheets





PSEUDOCODE: Represents an algorithm using statements.

Example

Draw a flowchart and pseudocode program to multiply two numbers

Solution

- Pseudocode

Step1: START

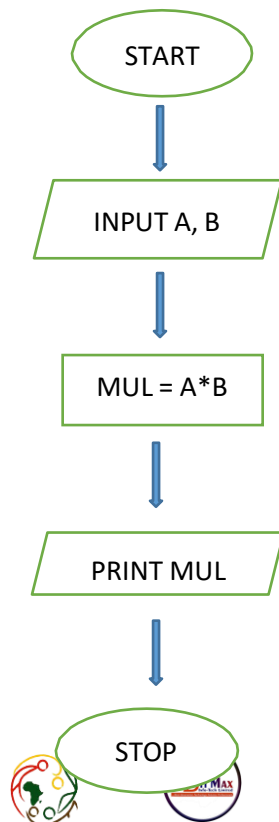
Step2: INPUT A, B

Step3: $MUL = A * B$

Step4: OUTPUT MUL

Step5: STOP

- Flowchart





2. Write a pseudocode and flowchart to accept a student's marks in Physics, Chemistry and Biology. The total of these marks as well as the average of should be displayed.

ANS

Pseudocode

START

DISPLAY "INPUT Physics mark"

INPUT a

DISPLAY "INPUT Chemistry mark"

INPUT b

DISPLAY "INPUT Biology mark"

INPUT c

Total = a + b + c

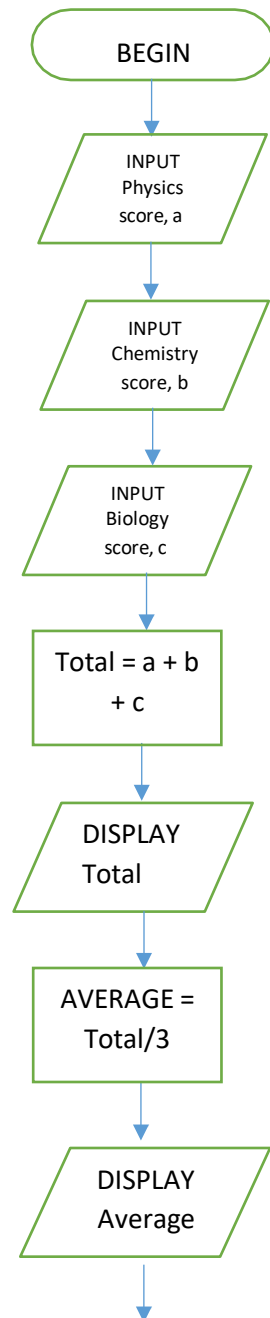
DISPLAY Total

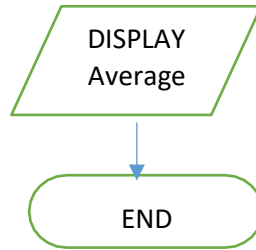
Average = Total/3

DISPLAY Average

STOP







CHARACTERISTIC OF PROGRAMMING LANGUAGE

1. It consist of Alphabets and alphanumeric numbers
2. It has a structure that consists of keywords, expressions and statements
3. It has syntax rules
4. It is processed by computer
5. It may require a translator

TRANSLATOR: A translator is a program that translate programs written in Assembly or High-level language into Machine language.

Examples of Translators are: Compilers, Assemblers and Interpreters

Compiler: A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. i.e. (0's and 1's)

Assemblers: An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.

Interpreter: an interpreter is a computer program that directly executes instructions written in a programming or scripting language, without requiring them previously to have been compiled into a machine language program.





LEVELS OF PROGRAMMING LANGUAGE

There are 3 levels of programming Language which are:

1. Machine language
2. Assembly Language
3. High-level Language

Low-Level or Machine Language

This is a programming language that provides little or no abstraction of programming concepts is very close to writing actual machine instructions. Two good examples of low-level languages are assembly and machine code.

High-Level Language

This a programming language such as c, Pascal, FORTRAN that enables a programmer to write programs that are more or less independent of a particular computer. Such languages are considered high level language because they are closer to human languages and further from machine languages. Examples of high-level languages are C, FORTAN or Pascal.

Assembly language

This is a low-level programming language designed for a specific type of processor. It may be produced by compiling source code from a high level programming language (such as C/C++) but can also be written from scratch. Examples include C, C++.

