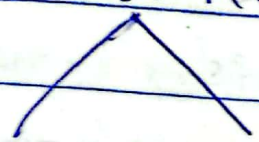
FORTRAN



Formula + Translation

Arithmetic, quantitative abilities aided FORTRAN assimilation when it was invented.

Your level of computer usage is based on your level of understanding of Low Level binary codes.

Low level languages are built on 0 1 binary codes.
They are ; — Assembly Language
— Machine Language

Packages are softwares written in a programming language For specific needs — Excel, Ms Word etc.

Packages were developed using languages; hence, languages are taught in schools; not Packages.

The Operating System is the software that enables the computer to start.

Alphabets are the finite collection or set of Symbols.

02 INTRODUCTION TO COMPUTER PROGRAMMING

(1). Program definition and analysis

(2). Decide an algorithm that can solve the problem

(a). An algorithm is a sequentially arranged set of instructions that can be mechanically followed to produce a result.

(3). Draw a flow chart

(4). Choose a language ⇒ languages are problem specific

(5). Start coding

(a). Coding means translating the language you used in the program

(6). Move to the keyboard

(7). Compilation and Debugging

Compilation : process of translating a program, written in Programming language into machine language

Debugging : process of correcting errors

(8). The program must be tested to be error free and capable of producing result, ie Execution

(9). The post execution state

CLASSIFICATION OF ERRORS

In programming, there are 3 classes of Errors

(1). Syntax or Compilation Error

(2). Execution Error (3). Logical or Danger Error

(c). SYNTAX ERROR

Occur as a result of wrong use of the syntax (grammar or rules) of the language of implementation.

(d). EXECUTION ERROR OR RUN-TIME ERRORS

Occurs as a result of mismash; a case where a computer is instructed to do something impossible. Eg

- x divided by 0
- Data mismatch
- Referencing an invalid file

(3). LOGICAL ERROR

Occurs as a result of wrong instructions.

15-05-2024

QUALITIES OF A GOOD PROGRAM

(a). The program:

(1). must be easy to understand, by the user and the designer

(2) must produce a correct result

(3) must be easy to maintain and update when necessary

(4) must be efficient

Program efficiency is determined by 2 things:

(a). Time used to process work

(b). Amount of storage required to run

A program that uses less time and storage is said to be efficient.

(5). Must be reliable, ie, it must have a check against unforseen circumstances (check against: handle, provide an alternative).

(6). Must be flexible ie the program must not be rigid or too strict

(7). Must be portable ie machine friendly (capable of functioning in all machines), social or can adapt easily.

A necessary approach to achieving all of these qualities is by the use of structured programming concept and with a considerable use of programming aids.

PROGRAM DESIGN TOOLS AND AIDS

(1). Structured Chart
(2). Flow Chart
(3). Pseudo Code
(4). Decision table
(5). Decision Tree
(6). HIPO Chart
(7). Visual Table of Content
(8). Data flow diagram (Charts)
(9). ✓ Structured diagram

ESSENTIAL COMPONENTS OF PROGRAMMING LANGUAGE

(1). INPUT / OUTPUT STATEMENTS

Statements that allows the computer to enter data and release results.

(2). ARITHMETIC STATEMENTS

Also called Data Manipulation and Processing Statements

Data + Processing = Information

(3). LOGIC STATEMENTS

Statements that helps you activate or ignite logic (which is one of the ways we communicate).

(4). CONTROL STATEMENTS

Statements that enables you to move data from one storage to another

(5). DATA MOVEMENT TRANSFER

(6). Statements used to inform the computer of the contents of a particular folder or space, using variable names; so the computer can ascertain the memory size needed for the supposed content.

SPECIFICATION STATEMENT

It is also called Declaration Statements.

CONTROL STRUCTURE

A control structure is a programming terminology that

determines the order of execution of a program.

A program is a finite collection of statements, written in a specific language.

TYPES OF CONTROL STRUCTURES THAT CAN BE APPLIED TO PROGRAMS

(1). Unconditional Control Structure
(2) Sequential ✓ ✓
(3). Selection ✓ ✓
(4). Repetition ✓ ✓

FLOW CHART(S).

Flow charts are diagramatic representations of logical steps involved in providing a solution to a typical problem.

03 FORTRAN CHARACTERS

May 20, 2024

When learning a new Language, it is important to first learn the following: The Alphabets, First 10 digits (Numbers) and Special characters (13), which include; Blank, Equal, Plus, Minus, Asterisk, Left parenthesis, Right parenthesis, Slash, Comma, Period, Currency Symbol, Apostrophe, Colon.

WHAT IS A STATEMENT?

A statement is an instruction that is built up by the combination of any of the characters in an alphabet.

Statements can be likened to a word in English Language. One doesn't need Complete knowledge of all statements to write a program or language.

WHAT IS A PROGRAM

A program is a carefully arranged sequence of statement that tells the Computer what to do.

A program is a carefully arranged sequence of statements which when executed, will be able to translate your data into information or result.

VARIABLES

When you write a program using data, those data are used to yield result. Those data written need to be stored into the computer.

Variable names are the names used when storing these data into the computer.

Variable in FORTRAN could be One of these six (6) types!

(1). Integer (2). Real (3). Complex

(4). Character (5). Logical Boolean (6). Double-precision

EXPRESSIONS AND OPERATORS

An expression is a combination of one or more constants, using an Operator.

Examples of Expressions

A + B

Operand | Operand

Operator

An expression can be ; Logical, Lexical or Arithmetic

— When an expression is evaluated, it will equally give you a value or a result.

Operators include;

(1). Arithmetic Operators

(a). ( ) bracket (b). ** Exponentiation (c). * Multiplication

(d). / Division (e). + plus (f). — Minus (g). = Assign to

(2). Relational Operators

(a). EQ Equal to (b). NE Not equal to (c). LT Less than (d). LE Less than or Equal to

(e). GE Greater than or Equal to (f). GT Greater than.

(3). Logical Operators

(a). And (both of) (b). EQV

(c). Or (d). not

FUNCTIONS

FORTRAN as a language has its own compiler. Within the compiler, are In-built or Intrinsic Functions.

These functions will evaluate mathematical functions commonly used or needed. They include; sine, cosine, Log etc.

Q4 ESSENTIALS OF FORTRAN LANGUAGE

TYPES OF STATEMENTS

(1). PROGRAM STATEMENT

Program statement is a non-executable statement with the general form (syntax)

PROGRAM name

Where name is the name of the main program. Its use in a program is optional and in cases where it is used, it must appear as the first line (statement) of any program.

(2). COMMENT STATEMENTS

— It is a non-executable statement

Its general form is C or *

— these statements are not meant for the computer to execute. It is meant for the user, as it will aid the user's understanding of the program.

(3). ASSIGNMENT STATEMENTS

Its general syntax is

Variable name = Expression

The expression consists of combination of variables, constants and operators

A = 4 * B + C

variable name — expression

It is important to let the computer know the value to be stored so memory usage can be optimised. This is made possible by informing the computer of the value to be stored in the variable name.

(4). TYPE DECLARATION STATEMENTS

- Statements used to tell the computer the type of Data you intend to store under any of your variable names.

The general syntax of the commonly used Type-declaration statements are:

(a). INTEGER list of variables (separated by comma if more than one)

(b). REAL list of variable

(c). DOUBLE-PRECISION list of variables

(d). COMPLEX list of variables

(e). LOGICAL list of variables

(f). CHARACTER *n list of variables

⟹ Multiple declaration of a variable will produce an error during program

(5). PARAMETER STATEMENTS

- Statement used to define the names of constants used often within a program.

Its general syntax is

Variable 1 = value 1; variable 2 = value 2 .... Variable n = value n.

PARAMETER (pie = 3.142, G = 9.8, T = 34)

(6) DATA STATEMENTS

- Unlike the parameter statement, this statement is used to assign initial variables values to variables.

Its general syntax

DATA use of variables / List of values

⟹ Data, Parameter & Assignment statements are used to assign values to variable names.

DATA A, B, C / 3, 4, 5, 6, 12

(7). STOP STATEMENT

— An executable statement marking the end of all executable statements in a program.

— It is used to tell the computer to stop execution (under abnormal conditions).

Its general syntax is **STOP**

(8). END STATEMENT

This statement is closely related with stop.

— This marks the final end of a program or subprogram (s) used within the main program.

— Its general syntax is **END**

(9). OPEN STATEMENT

Results from the execution of a program are usually saved into a file. Each of these two files [INPUT / OUTPUT] need to be specified within the main program. This statement is used to assign unit number (n_1) to a file with the name Fn.

(10) READ STATEMENT

— Besides the use of parameter, assignment and Data statements, a major

and structured approach of input is through the use of READ & FORMAT Statements in FORTRAN.

A simple syntax for Read statement is of the form:

READ (unit number, Format-label) list

Eg: READ (5, 100) A, B, C

This is instructing the computer to assign values to the variables A, B, C from a file with unit number 5, using the data layout specification given in FORMAT statement with label 100.

(ii) WRITE STATEMENT

Output of results of program execution is accomplished through the use of a WRITE/PRINT statement.

Its general syntax is: WRITE (unit number, Format-label) list

This statement enables you to write rather than read.

Eg: 2A10

Which means reserve 2 fields made up of 10 spaces each.

(ii). FORMAT STATEMENTS

The physical layout of data and information in both your input & output files are specified using a Format statement.

Its general form is:

Label FORMAT (specification)

Field specification is the general layout of information.

FIELD DESCRIPTION(S).

A field is a space reserved for a unit of information in a file.

⟹ A record is a collection of related fields.

(1). I-FIELD DESCRIPTOR (rIm)

- Used to specify an integer location

Where r is the number of times the field repeats

✓ m is a positive integer indicating the width of the field.

(2). F-FIELD DESCRIPTOR (rFm.n)

- Used for specify general layout for real variables

(3). X SPECIFICATION (rX)

- Used to allocate free space within the data file.

(4). A-FIELD DESCRIPTOR (rAm)

- Used to specify the space reserved for character (Alpha numeric) values

Eg: matriculation number

(5). E-FIELD DESCRIPTOR (rEm.n)

- Used to describe spaces to be used for storing real numbers, which will be stored in exponential formats.

(6). G-FIELD DESCRIPTOR (rGm.n)

- Used to specify a real field where the data's storage is determined based on the computer's discretion.

(7). D-FIELD DESCRIPTOR (rDm.n)

- Used to store numbers that have double-precision ie numbers with

negligible values which will eventually have great significance.

(8). TAB - FIELD DESCRIPTOR (Tm)

- It tells in what column an output field defined in the descriptor succeeding the tab should begin.

(9). L - FIELD DESCRIPTOR (rLm)

- used to read or write logical variables

(10). SLASH (/)

- Used for creating lines.