UNIT I (GE8151 PROBLEM SOLVING AND PYTHON PROGRAMMING)

NOTATION

Pseudo code

Pseudo code is made up of two words: Pseudo and code. Pseudo means 'imitation' and 'code' refers to instructions written in programming language. Pseudo code is not a real programming language, but it looks like a programming language. Pseudo code is also called as "Program Design Language [PDL]". It is an outline of a program, written in a form that can be easily converted into real programming statements. Pseudo code instructions are written in normal English.

Rules for writing pseudo code:

i) Write one statement per line.

ii) Capitalize the keywords.

iii) End Multiline structure.

iv) Keep Statements language independent.

v) Intend to show hierarchy.

Keywords used in pseudo code:

START: BEGIN

INPUT: READ, OBTAIN, GET, INPUT, DEFINE

OUTPUT: OUTPUT, PRINT, DISPLAY, SHOW

COMPUTE:CALCULATE, COMPUTE, ADD, SUBTRACT, INITIALISE, DETERMINE

INITIALIZE: SET, INITIALIZE

ADD ONE: INCREMENT

STOP: END

Pseudo code guidelines:

i) Pseudo code statements should be written in simple English.

ii) Each statement should be written in separate line.

iii) The keywords should be capitalized.

iv) Pseudo code should be programming language independent.

v) The steps must be understandable.

vi) Each set of instructions are written from top to bottom.

Advantages (Benefits):

⟶ It can be read and understood easily.

⟶ It can be done easily on a word processor.

⟶ It can be modified easily.

⟶ It occupies less space.

⟶ It will not run over many pages.

⟶ Converting a pseudo code to a program is simple.

Disadvantages (Limitations):

⟶ It is not visual**.**

⟶ We do not get a picture of the design.

⟶ There is no standardized style or format.

⟶ For a beginner, it is more difficult to follow the logic or write pseudo code.

Example 1: Write Pseudo code to calculate sum and average for n numbers.

```
BEGIN
INITIALIZE sum=0, i=1
READ n
FOR i < = n, then
        COMPUTE sum = sum +i
        CALCULATE i=i+1
END FOR
COMPUTE avg = sum/n
PRINT sum, avg
END
```

Example 2: Write Pseudo code to add two numbers**.**

```
BEGIN
SET C=0
READ A, B
ADD C=A+B
PRINT C
```

```
END
```

Example 3: Write Pseudo code to calculate area of circle.

```
BEGIN
READ radius r
INITIALIZE pi=3.14
CALCULATE Area=pi * r *r
PRINT Area
END
```

Example 4: Write Pseudo code to read number n and print the integers counting up to n.

```
BEGIN
READ n
INITIALIZE i to 1
FOR i <= n, then
        DISPLAY i
        INCREMENT i
END FOR
END
```

Example 5: Write Pseudo code to find the greatest among two numbers.

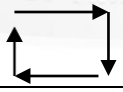
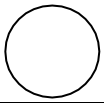
```
BEGIN
Read A, B
IF A >B
        PRINT "A is greatest"
ELSE

        PRINT "B is greatest"
ENDIF
END
```

Flowchart

Flowchart is a diagrammatic representation of an algorithm. A flowchart is a picture of the separate steps of a process in sequential order. Flowchart is made up of boxes, diamonds and other shapes connected by arrows where each shape represents a step in the process. The arrows show the order of the flow of execution. Flowcharts are used in designing or documenting a process or program. The logic of the program is communicated in a much better way by using a flowchart.

Flowchart symbols:

Sl.No	Name of the symbol	Symbol	Description
1	Start/Stop		Represent the start and stop of the program
2	Input/Output		Denoted either an input or output operation
3	Process		Denotes the process to be carried out
4	Decision		Represent decision making and branching
5	Flow lines		Represents the sequence of steps and direction of flow
6	Connector		Connects remote parts of the flowchart on the same page

Guidelines for drawing flowchart:

i) All necessary requirements should be listed out in logical order.

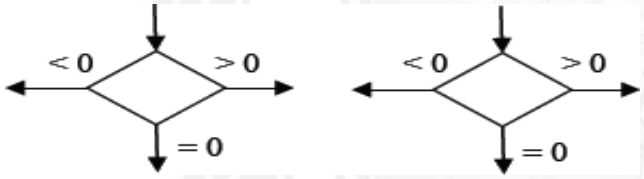
ii) There should be **START** and **STOP** in the flowchart.

iii) The flowchart should be clear, neat and easy to follow.

iv) The direction of flow is from left to right or top to bottom.

v) Only one flow line should emerge from a process symbol.

vi) Only one flow line should enter a decision symbol but 2 or 3 flow line can leave the decision symbol.



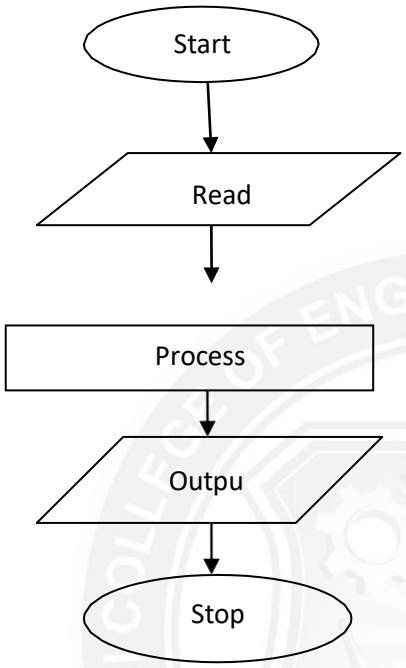
vii) Only one flow line is used with terminal symbol.

viii) If the flowchart becomes complex, connector symbols are used to reduce the number of flow lines.

ix) The text within the symbols should be brief.

x) The validity of flowchart should be tested by passing simple test data.

Basic design structure of flowchart:



Advantages:

i) Better communication

It is easy for the programmer to explain the logic of program.

ii) Effective analysis

It is easy to analyze the problem effectively.

iii) Proper documentation

With the help of flowchart good documentation is done for various purposes.

iv) Efficient coding

Flowchart acts as a guide during the system analysis and program development phase.

v) Efficient debugging

It helps in debugging process.

vi) Efficient program maintenance

The maintenance of a program becomes easy with the help of the flowchart.

Disadvantages (Limitations):

i) Complex logic

Sometimes the logic of the program is quite complicated. In such a case flowcharts become complex.

ii) Alterations and modifications

If alterations are required, the flowchart needs to be redrawn completely.

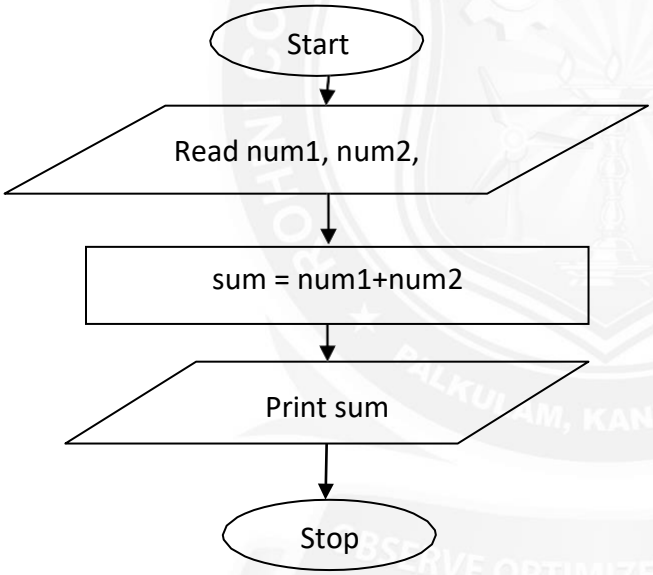
iii) Reproduction

Reproduction of the flowchart becomes a problem because it cannot be typed.

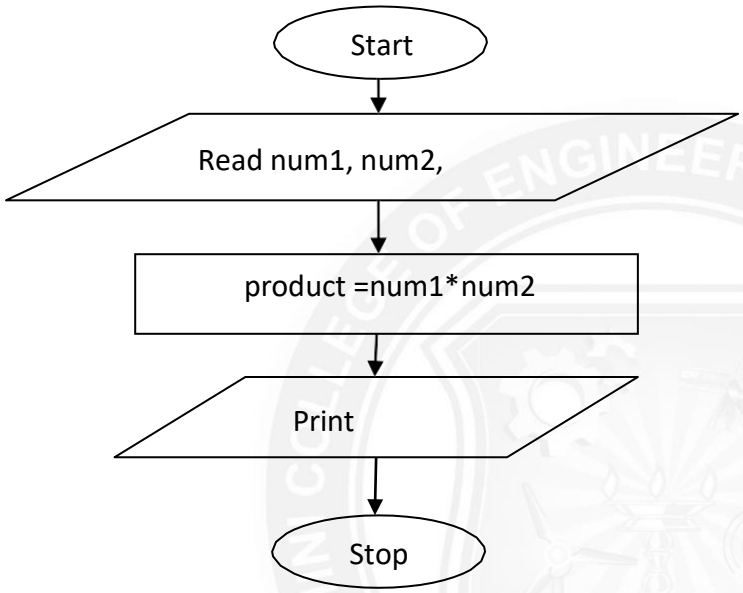
iv) Cost

High cost for large applications.

Example1: Draw a flowchart to add two numbers.



Example2: Draw a flowchart to find the product of two numbers.



ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY
PALKULAM, KANYAKUMARI
OBSERVE OPTIMIZE OUTSPREAD