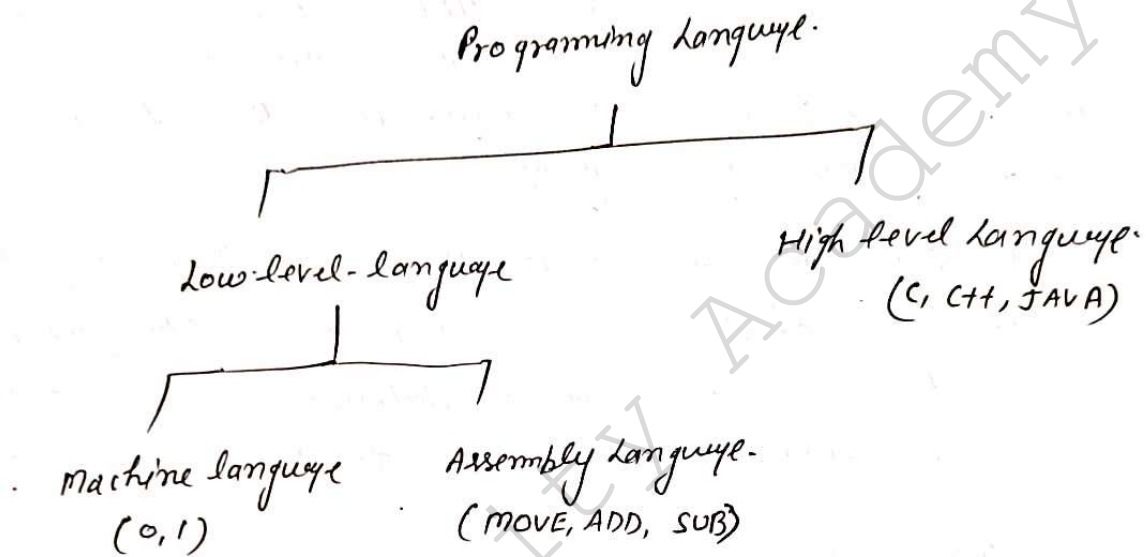


1.2.4. Concept of Assembler, Compiler, Interpreter, loader and linker.

- i) Concept of Programming Languages.
- ii) Concept of translators.
- iii) Concept of loader and linker.

i) Concept of Programming Language.



- a) Machine Language: Machine language is composed of (0,1) binary digit. Machine language is only language a computer is capable to understand.
- b) Assembly language: Assembly language uses alpha-Numeric code instead of binary digit. It is easy to remember than machine language. Assembly language is machine dependent language.
- c) High level language: It is machine independent lang. it is easy to read, write and maintain as it is written in english like words. HLL is portable.

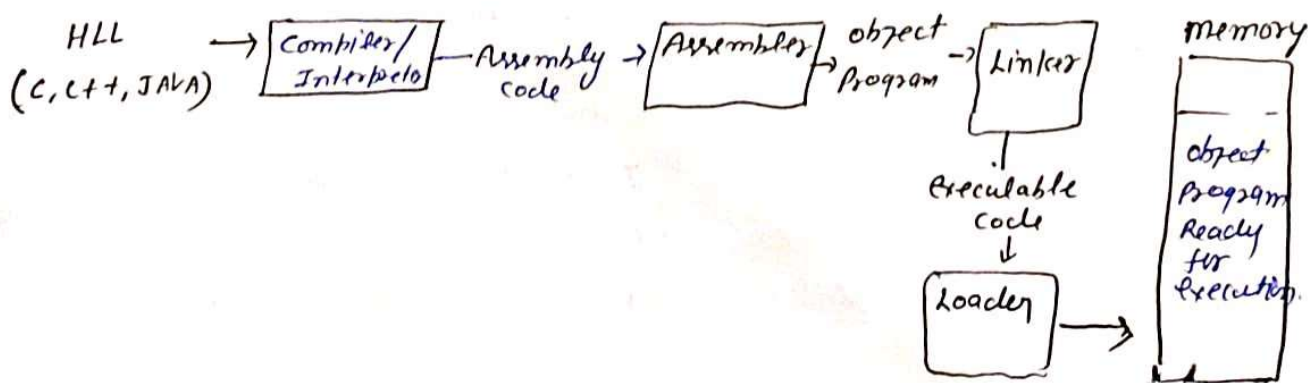
2) Concept of Translator: The translators are just computer programs which accept a program written in high level or low-level language and produce equivalent machine language as output. There are three translators.

- **Assembler:** used to convert Assembly code into machine language.
- **Compiler:** used to convert HLL to machine language. Compiler searches all the errors of program and lists them. If the program is error free then it converts code into machine language.
- **Interpreter:** used to convert HLL to machine language. Interpreter checks the error of program statement by statement. After checking one statement it converts it into machine language and then executes that statement.

Note:

- Machine language → No need of translator
- Assembly language → Assembler used to convert Assembly to machine lang.
- High level language → Compiler or ~~Interpreter~~ Interpreter used to convert HLL to low level language.

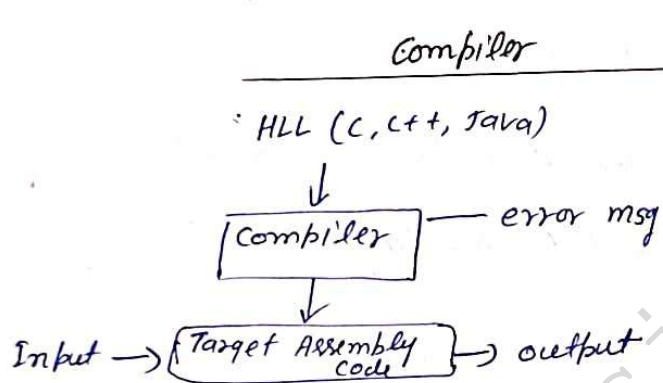
3) Concept of Linker and Loader.



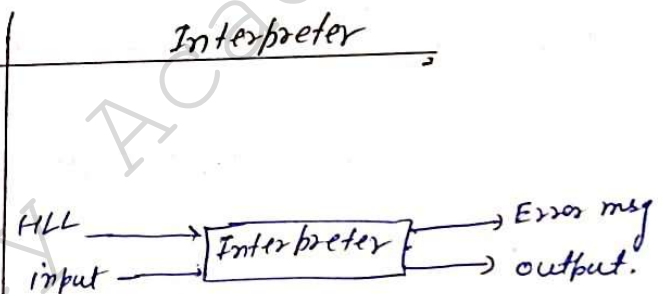
Linker: A linker combine object file generated compiler into a single executable (.exe) file. A linker is also responsible to link and combine all the module of program if written separately.

Loader: A loader load the program into main memory from storage device. the operating system calls the loader when needed.

Difference between Compiler and Interpreter.



- 1- Compiler translate source code into object code as whole.
- 2- It create object file.
- 3- Execution is fast
- 4- Program not require to ~~run~~ translate each time to run the program.
- 5- Not easy to correct error
- 6- Most HLL uses compiler. eg.
C, C++, FORTRAN



1. Interpreter translate statements of source code one by one and execute immediately.
2. It does not create object file.
3. Execution is slow.
- 4- Program require to translate each time to run.
- 5- Easy to correct mistake in source code.
- 6- few languages uses Interpreter eg. lisp, Python, Basic. etc.

1.3. Idea of Algorithm.

The word algorithm comes from the name of a Persian author, Abu Ja'far who wrote a text book on mathematics. Algorithm has come to refer to a method that can be used by computer for the solution of problem.

An algorithm is a finite set of instructions that to solve any problem. Every algorithm must follow following criteria:

1. Input : zero or more.
2. Output : atleast one output is produced
3. Definiteness : Each Instruction must be clear and unambiguous.
4. Effectiveness : Each algorithm must be produce effective output as desired.
5. Finiteness : Each algorithm must be terminate after finite number of steps.

1.3.1 Representation of Algorithm.

Example.

Write Algorithm to Add two number

1. start
2. input A, B
3. calculate sum = $A+B$
4. Display sum
5. stop.

Steps to write Algorithm.

- 1- start.
2. Input
3. Processing / calculation
- 4- output
- 5- stop.

Example 2. write An Algorithm to calculate Average of 5 numbers

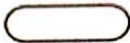

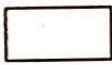

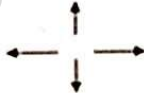




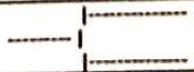
- 1- Start
2. Input A, B, C, D, E
- 3 calculate $SUM = A + B + C + D + E$
- 4 calculate $Avg = SUM / 5$
- 5- Display Avg.
6. Stop.

Example 3. write an algorithm to find largest among three different number:

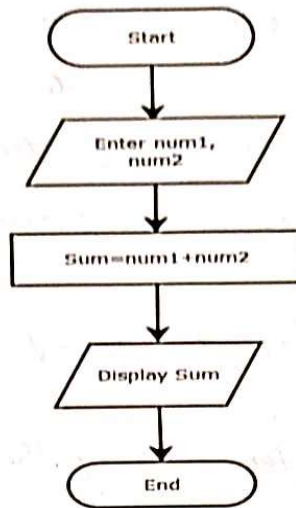
1. Start
2. Input A, B, C
3. if $A > B$
 - if $A > C$
 - Display A is largest
 - else
 - Display C is largest
- else
 - if $B > C$
 - Display B is largest
 - else
 - Display C is largest.
- 4- Stop.

1.3.2. Flow chart

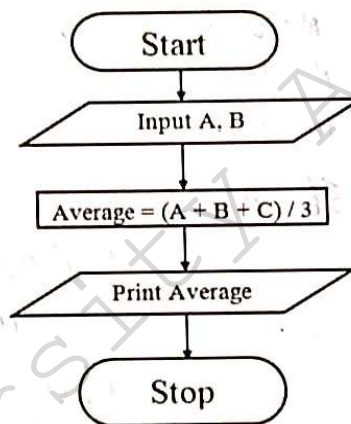
A pictorial representation of an algorithm is called flow chart. In flow chart the steps in the algorithm are represented in the form of different shapes, for example.

Symbol Name	Symbol	function
Oval		Used to represent start and end of flowchart
Parallelogram		Used for input and output operation
Rectangle		Processing: Used for arithmetic operations and data-manipulations
Diamond		Decision making. Used to represent the operation in which there are two/three alternatives, true and false etc
Arrows		Flow line Used to indicate the flow of logic by connecting symbols
Circle		Page Connector
		Off Page Connector
		Predefined Process /Function Used to represent a group of statements performing one processing task.
		Preprocessor
		Comments

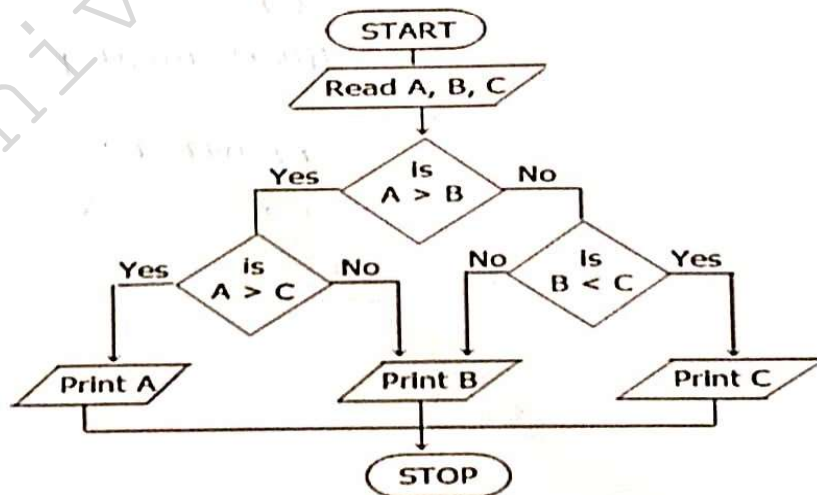
Example 1: Draw the flow chart to add two numbers



Example 2: Draw the flow chart to find average of Three numbers



Example 3: Draw flowchart to find largest of three numbers



1.3.3 Pseudo Code

Pseudo Code is a combination of two words Pseudo and Code. Pseudo means imitation and code refers to instruction. Pseudo code is not a real programming code. It is general way of describing an algorithm without using any specific programming language-related notations.

Pseudo Code is text based detail design tool.

Example: Pseudo code for finding the largest of three numbers

PSEUDOCODE BiggerOfThree:

Read A;

Read B;

Read C;

IF (A > B)

THEN IF (A > C)

THEN Print A;

ELSE Print C;

END IF;

ELSE IF (B > C)

THEN Print B

ELSE PRINT C;

ENDIF;

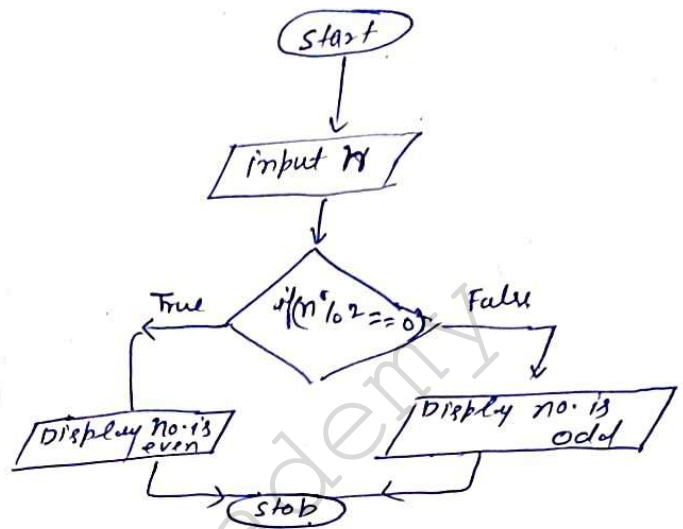
ENDIF;

END.

Exercise.

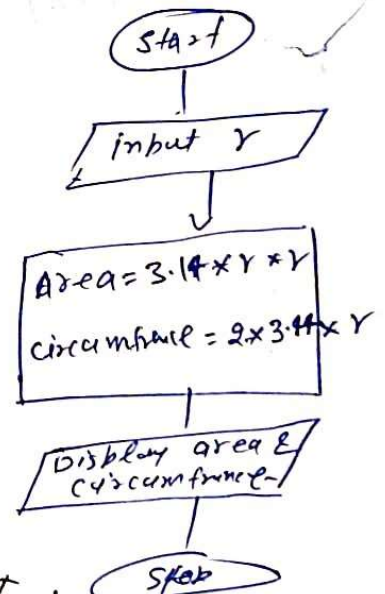
1. Write an algorithm and draw flow chart to check number is even or odd.

1. Start
2. input n
3. if $(n \% 2 \neq 0)$
 Display number even
else
 Display number is odd
4. stop



2. Write an algorithm and draw flow chart to calculate area and circumference of circle.

1. Start
2. input r
3. Calculate area $= 3.14 \times r \times r$
4. Calculate circumference $= 2 \times 3.14 \times r$
5. Display area and circumference.
6. Stop.

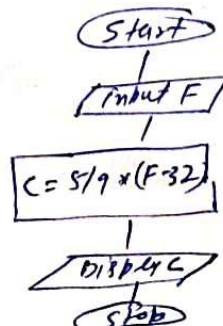


3. Write an algo and draw flow chart to convert temp F into C .

$$C = 5/9 \times (F - 32)$$

Algo

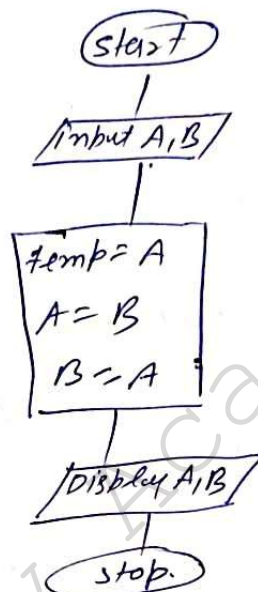
1. Start
2. input F
3. $C = 5/9 \times (F - 32)$
4. Display C
5. Stop.



4. write an algorithm and draw flow chart to swapping of two number using third variable and without using third variable.

(i) using third variable.

1. Start
2. input A, B
3. temp = A
A = B
B = temp
4. Display A, B.
5. stop



(ii) without using third variable.

1. Start
2. input A, B
3. $A = A + B$
 $B = A - B$
 $A = A - B$
4. Display A, B
5. stop.

