Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

RTMENT OF INFORMATION TECHNOLOGY

ROGRAMMING FOR PROBLEM SOLVING

I YEAR - I SEM

T 1 – Introduction to Problem Solving Techniques

tion [pseudo code, flow chart, and programming languag

can be expressed in many different notations, including Na
de, flowcharts and programming languages.
nguage tends to be verbose and ambiguous.
le and flowcharts are represented through structured human langua
is a system of characters, expressions, graphics or symbols desi
rs in problem solving to represent technical facts, created to fa
program
words Notations collectively represents the following:
lo code
charts
amming languages.

e is an informal high-level description of the operating principl
r algorithm.

basic structure of a normal programming language, but is inte
her than machine reading.

ised detail design tool.

ans 'false' and code refers to 'instructions' written in programmin

e cannot be compiled nor executed, and there are no real form

ocode is written in normal English language which cannot be u

code: To find sum of two numbers

```
EAD num1,num2
um=num1+num2
RINT sum
```

ent per line.
epresents single action is written on same line.
e to read the input, all the inputs must be read
statement.
al keywords
ds should be written in capital letters.
WRITE, IF, ELSE, ENDIF, WHILE, REPEAT
hierarchy
is a process of showing the boundaries of the

tructures
re must be ended properly, which provides more

language independent.
must never written or use any syntax of any
g language.

Example: 01

Pseudocode: Find the
subjects

```
READ name, mark1, m
Total=mark1+mark2+m
Average=Total/3
WRITE name, mark1, 
```

Example: 02

Pseudocode: Find great

```
READ a, b
IF a>b then
        PRINT a is grea
ELSE
        PRINT b is grea
ENDIF
```

s of Pseudocode
e done easily on a word processor
 modified
ments structured concepts well
be written easily
be read and understood easily
rting pseudocode to programming language is easy as co
hart

ages of Pseudocode
ot visual
is no standardized style or format

yword used to represent a comment.

ND: Begin is the first statement and end is the last staten

ET, READ: The keyword is used to inputting data.

E, CALCULATE: used for calculation of the resul

BTRACT, INITIALIZE: used for addition, subtraction an

PRINT, DISPLAY: It is used to display the output of the

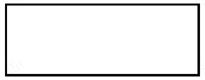
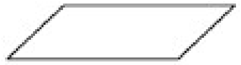
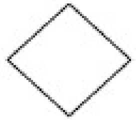
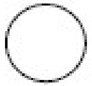
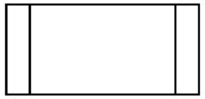
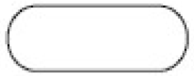
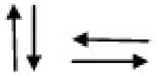
ENDIF: used to make decision.

ENDWHILE: used for iterative statements.

DFOR: Another iterative incremented/decremented testec l representation of an algorithm.

s is a diagram made up of boxes, diamonds, and other shapes, co

e represents a step in process and arrows show the order in which

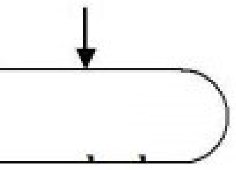
Symbol	Name	Function
	Process	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Terminal	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

Name	Symbol	Description
Process	▭	Process or action step
Flow line	→	Direction of process flow
Start/ terminator	⬭	Start or end point of process flow
Decision	◇	Represents a decision making point
Connector	○	Inspection point
Inventory	▽	Raw material storage
Inventory	△	Finished goods storage
Preparation	⬡	Initial setup and other preparation steps before start of process flow
Alternate process	▢	Shows a flow which is an alternative to normal flow
Flow line(dashed)	⇢	Alternate flow direction of information flow

per flowchart, all necessary requirements should be listed out in logical order.
d be clear, neat and easy to follow. There should not be any room for ambiguity in under
ons of the flow of a procedure or system is from left to right or top to bottom.
ow line should come out from a process symbol.

line should enter a decision symbol, but two or three flow lines, one for each possi

ne is used in conjunction with terminal symbol.



omes complex, it is better to use connector symbols to reduce the
es.
chart has logical start and stop.

Flowchart:

rts are better way of communicating the logic of the system.
is
help of flowchart, a problem can be analyzed in more effective way.
tation
rts are used for good program documentation, which is needed for various purposes.

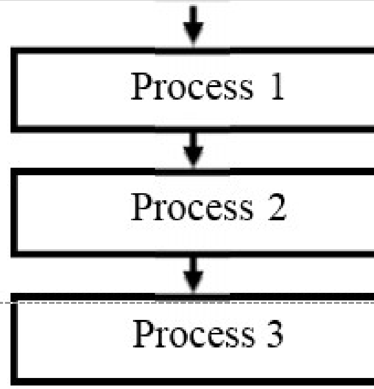
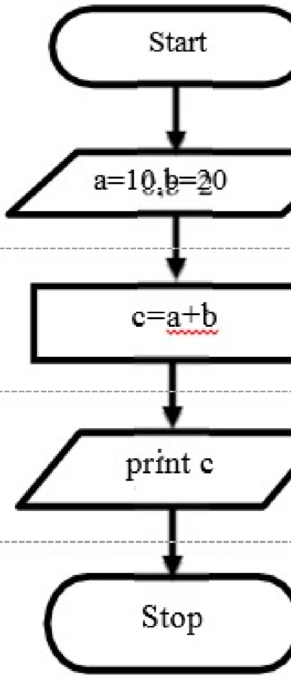
charts act as a guide or blue print during the system analysis and program development
ng and Debugging
chart helps in testing and debugging the program
n Maintenance
tenance of operating program becomes easy with the help of flowchart.
ne programmer to put efforts more efficiently on that part.
of Flowchart

es, the program logic is quite complicated. In that case flowchart becomes complex and
Iodification:
ons are required the flowchart may require redrawing completely.
s the flowchart symbols cannot be typed, reproduction becomes problematic.

PSEUDOCODE

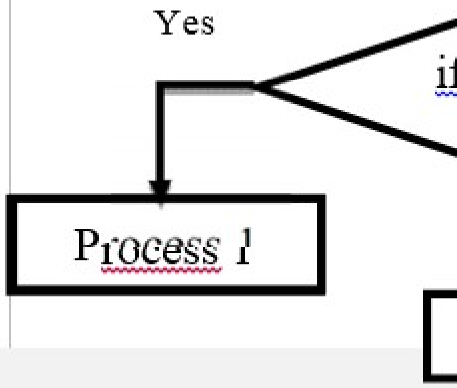
ucture

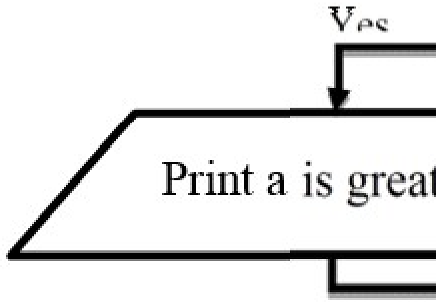
a series of steps that take
r another.
epresented here by a new line

Pseudocode	Flow Chart
General Structure	
Process 1 …. Process 2 … Process 3	
Example	
READ a READ b Result c=a+b PRINT c	

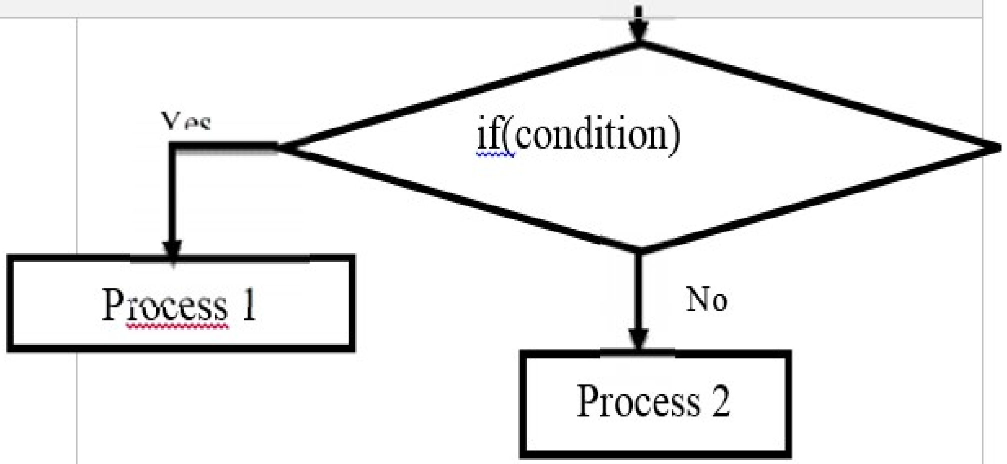
e is used to check the condition.
outputs only (True or False)
re the conditional structures used.

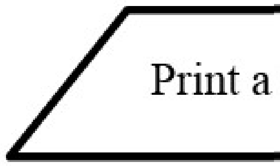
Pseudocode	**Flow Chart**
General Structure	

IF condition THEN
    Process 1
ENDIF
	

Example	
READ a READ b IF a>b THEN PRINT a is greater	

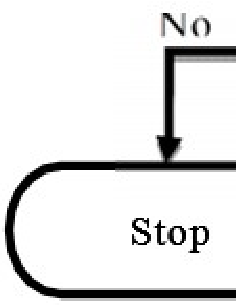
is the structure used to specify, if
, then execute Process1, else, that
then execute Process2

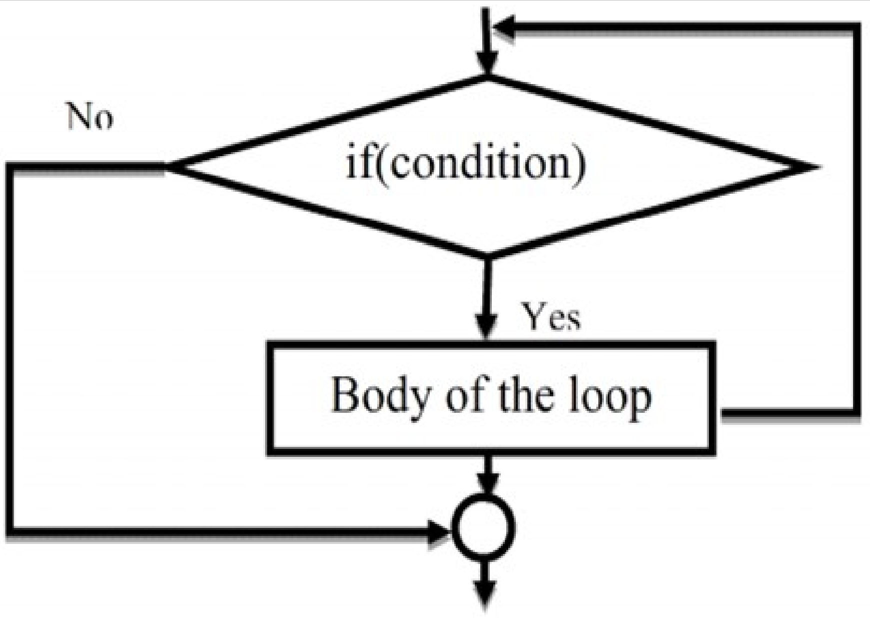
	Flow Chart
	

READ a READ b IF a>b THEN PRINT a is greater	

rally used with WHILE or
FOR loop.
R is <u>**entry checked loop.**</u>
. exit checked loop, so the
cuted at least once.

```
INITIALIZE a=1
WHILE a<10 THEN
        PRINT a
        a=a+1
ENDWHILE
```



	Flow Chart
	

	Flowchart	Pseudo cod
s a sequence used to m	It is a graphical representation of algorithm	It is a langua representati algorithm.
owledge to n.	not need knowledge of program to draw or understand flowchart	Not need kn program lan understand pseudo code

ming language is a vocabulary and set of grammatical rules
r computing device to perform specific tasks.
rd it is set of instructions for the computer to solve the prob
ng Language is a formal language with set of instruction, to
blem.
n will accept the data to perform computation.
mmers have to follow all the specified rules before writing p
ng language.
s to communicate with the computer using language which

Program= Algorithm + Data

ogramming Languages

amming languages are also used to organize the computatio

; Programming language we can solve different problems.

prove the efficiency of the programs.

ogramming Language

rogramming languages are classified into three types. They

– level or Machine Language

nediate or Assembly Language

– level Programming language

ge:

language is the lowest-level programming language.

languages are the only languages understood by computers.

called as low level language.

xample code:100110011

» 111001100

ee:

anguage is the only language which the computer understands.

ing any program written in any programming language, the conversion to machine langu

am written in machine language can be executed directly on computer.

e any conversion process is not required.

ne language program is translation free.

conversion time is saved, the execution of machine language program is extremely fast.

o find errors in a program written in the machine language.

ogram in machine language is a time consuming process.

nguage:

rcome the issues in programming language and make the programm

an assembly language is developed which is logically equivalent to

ge but it is easier for people to read, write and understand.

bly language is symbolic representation of machine language.

bly languages are symbolic programming language that uses symb

nt machine language instructions.

re called low level language because they are so closely related to

sembly language contains the same instructions as a machine langu

ctions and variables have names instead of being just numbers.

sembly language consists of mnemonics, mnemonics that correspo

ne instruction.

Example code: start

» Add x,y

» Sub x,y

e program which translates assembly language instruction in to a machine l
understand and use.
y to locate and correct errors.

dent:
embly language program which can be executed on the machine depends on
nputer.

chine dependent, so the programmer should have the hardware knowledge to
sembly language.

on time of assembly language program is more than machine language prog
assembler is needed to convert from assembly language to machine langua

nguage:

vel language contains English words and symbols.

cified rules are to be followed while writing program in high level language

erpreter or compilers are used for converting these programs in to machine

evel language (HLL) is a programming language such as C, FORTRAN, or

mmer to write programs that are more or less independent of a particular ty

guages are considered high-level because they are closer to human languag

languages.

ely, programs written in a high-level language must be translated into machi

r or interpreter.

xample code: print("Hello World!")

high level language to machine language:

rograms that translate high level language in to machine language are called

ler.

compiler is a program which translates the source code written in a high level language i
chine language program.
mpiler reads the whole program written in high level language and translates it to machi
ny error is found it display error message on the screen.

erpreter translates the high level language program in line by line manner.
e interpreter translates a high level language statement in a source program to a machine
mediately before translating the next statement.
hen an error is found the execution of the program is halted and error message is display

adability:

High level language is closer to natural language so they are easier to learn and underst

chine independent:

High level language program have the advantage of being portable between machines.

sy debugging:

Easy to find and correct error in high level language

ges:

ss efficient:

The translation process increases the execution time of the program.

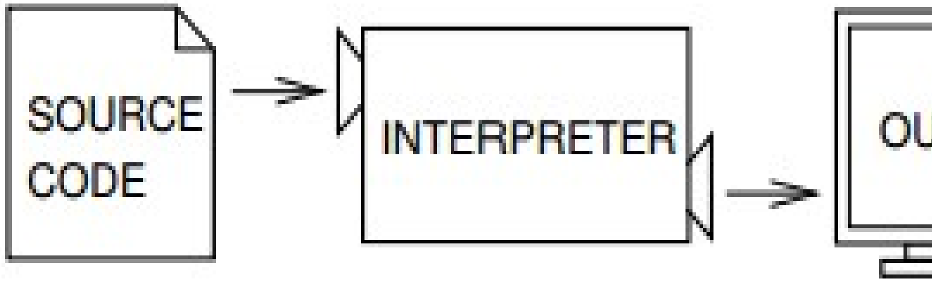
Programs in high level language require more memory and take more execution time to ramming languages are further divided and shown in the Table.

Type	Example
Programming Language	Python, BASIC, Lisp
Programming Language	Clean, Curry, F#
rogramming Language	C++,Java, Ada, ALGOL
Programming Language	C,Matlab, CList
ogramming Language	PHP,Apple Script, Javascri
gramming Language	HTML,SGML,XML
gramming Language	Prolog, Fril
Programming Language	ABCL, Concurrent PASCA
nted Programming Language	C++,Ada, Java, Python

rogramming Language:

eter is a program that executes instructions written in a high-level erpreter reads the source code one instruction or one line at a time, o a machine code and executes it.

cal, Python

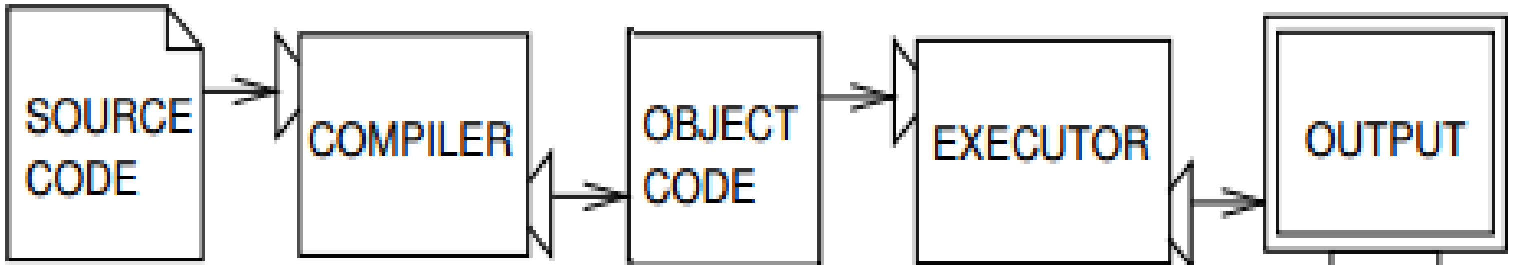


ogramming Language:

le is to transform a program written in a high-level programming l code into object code.

n be done by using a tool called compiler.

piler reads the whole source code and translates it into a complete n to perform the required tasks which is output as a new file. Ex: C



Programming Language	Compile Programming Langu
e statement at a time	Scans entire program and transla into machine code
mount of time to analyze the out the overall execution time is	It takes large amount of time to a source code but the overall execu comparatively faster
ate object code is generated, mory efficient	Generates intermediate object co further requires linking, hence re memory
nslating the program until first n which case it stops. Hence easy.	It generates the error message or scanning the whole program. He is comparatively hard.
Ruby	Eg: C,C++,Java