PYTHON : PROGRAMMING LANGUAGE FOR FUTURE!

Premanand S

MODULE 1 INTRODUCTION TO PROBLEM SOLVING

- Problem solving: Definition and Steps
- Problem Analysis Chart (PAC)
- Developing an Algorithm
- Flowchart
- Pseudocode

Premanand S

BASIC TERMINOLOGIES

TERMINOLOGIES NEED TO KNOW,

- Programmer – solves problem, writes codes for company
- User – one who uses code written by programmer
- Software – collection of instruction for computer to perform some task
- Computer only knows, how to follow codes in software
- Programmer needs to know how to solve the problem

PROBLEM

PROBLEM ?

- What is problem?

- Humans – Problem
- No problem, no living things
- No limitations, no age

- Humans – Data's - Machines / System
- MNCs / Software / Clients / Job

PROBLEM SOLVING: DEFINITION & STEPS

- Computer technology – completing any task fast and accuracy
- Steps with skills
- Technology – not advanced enough to solve problem by own, hence we need to give step by step instruction to proceed
- Eg: Zomato, Swiggy – Ordering our favourite, Bookmyshow – Booking cinema in no time, Amazon, Netflix, Hotstar – Entertainment all in one place.
- Eg: Groceries in Netflix

“

EFFICIENCY IN SOLVING PROBLEM DEPENDS ON HOW CORRECTLY & PRECISELY WE DEFINE PROBLEM, DESIGNING THE SOLUTION (ALGORITHM) AND IMPLEMENTING THE SOLUTION (PROGRAM) USING PROGRAMMING LANGUAGE”

HOW PROBLEM CAN BE SOLVED BY COMPUTERS?

PROGRAM DEVELOPMENT CYCLE

- Analyze – Problem definition
- Design – Problem solution
- Choose reciepe – Flowchart, Pseudocode, Charts and algorithms
- Code – Converting algorithm to programming language
- Test & Debug – Real time errors
- Documentation – Problem solving to product development

STEPS INVOLVED FOR PROBLEM SOLVING,

- Solving problem by computers,
  - . Pre-programming phase
  - . Programming phase (from module 2)

PRE-PROGRAMMING PHASE

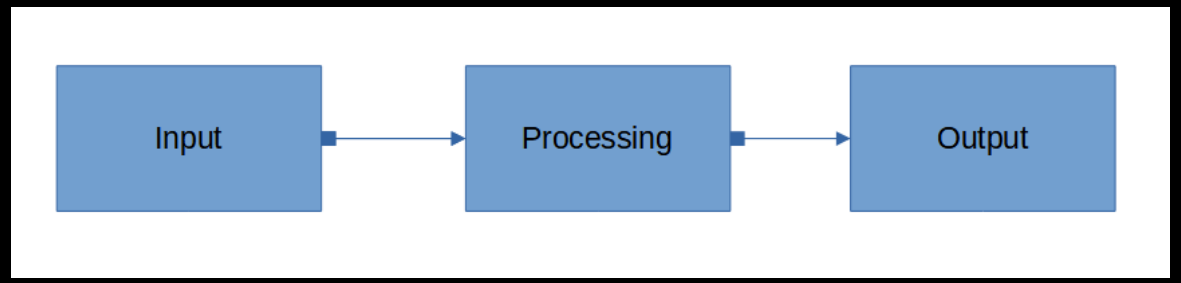
STEPS IN PRE-PROGRAMMING PHASE

- Analyzing the Problem – Problem Analyzing Chart (PAC) - beginning analysis of the problem
- Developing Interactivity Chart (IC) - overall layout or structure of the solution.
- Developing Input Process Output (IPO) chart - shows the input, the processing, and the output
- Algorithms - show the sequence of instructions comprising the solution
- Program Flowchart - which are graphic representations of the algorithms
- Pseudocode - represents a language like solution

ANALYZING THE PROBLEM – PROBLEM ANALYZING CHART (PAC)

ANALYZING THE PROBLEM

- Understand & Analyze problems – determine whether it can solved by computer
- Analyze the requirement of problem



PROBLEM ANALYSIS CHART (PAC)

- First step in solving a problem is analysing
- Good way to analyze the problem is to view in four parts

PROBLEM ANALYSIS CHART (PAC)

Given Data	Required result
Data given in the problem/ provided by the user. It can be any data or values or quantity	Requirements for the output reports including the information needed and the format required
Processing Required	**Solution Alternative**
List of processing required including equations or other types of processing.	List of ideas for the solution of the problem

EXAMPLE 1 FOR PAC

- Problem of being hungry

Given Data	Required Result	Processing	Solution Alternative
Food, Water, Juice	Tummy full	Food Preparation	Either eat on your own Feed by others

EXAMPLE 2 FOR PAC

- Calculate the amount for call driver for our own car, who works on hourly basis.

Given Data	Required Result	Processing	Solution Alternative
Hours worked and Pay rate	Amount in total	Amount = Hours work * Pay rate	Within city limit 1st three hours 300rs, each 1hour 60rs. Out of city limit 1st three hours 500rs, each 1hour 80rs

EXAMPLE 3 FOR PAC

- Calculate the amount needed for 2 days trip along with fuel consumptions among with friends.

Given Data	Required Result	Processing	Solution Alternative
Food, Snacks, Petrol and accommodations	Amount per head = Total amount / no.of persons	Accommodation per night with extra bed Snacks, food details Mileage of our car = kms run / amount of fuel used	One person is paying Each person is paying in each scenario

ASSIGNMENTS FOR PAC

- Calculate average of five numbers by using PAC
- Calculate the gross pay of an employee by using PAC
- Convert the distance in kilometres to miles where, 1mile = 1.609km by using PAC
- Calculate the gross pay of an employee given the hours worked and rate of pay. The gross pay is calculated by multiplying the hours worked by the rate of pay.
- Analyse the problem for area of circle, area = pi*radius*radius
- Write a PAC to compute and display the temperature inside the earth in Celsius & Fahrenheit, where Celsius = 10*(depth) + 20 & Fahrenheit = 1.8 * (Celsius) + 32

Premanand S

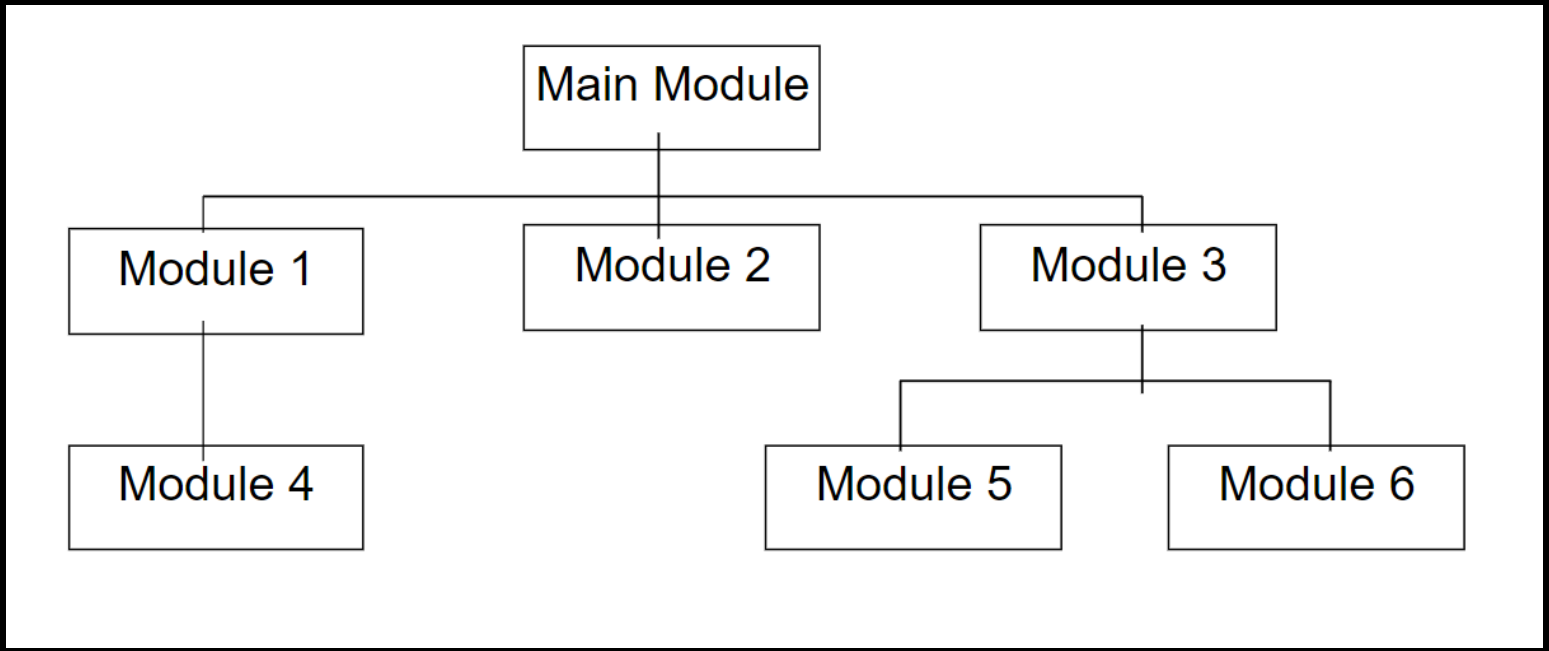
DEVELOPING INTERACTIVITY CHART (IC)

STEP 2

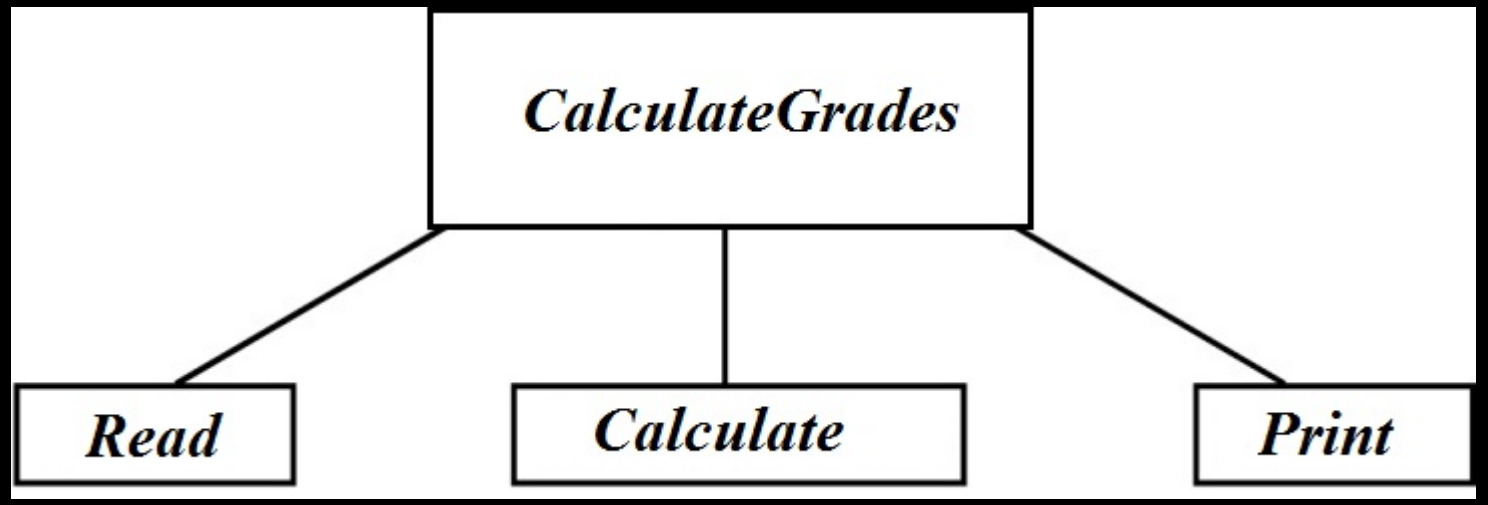
DEVELOPING IC

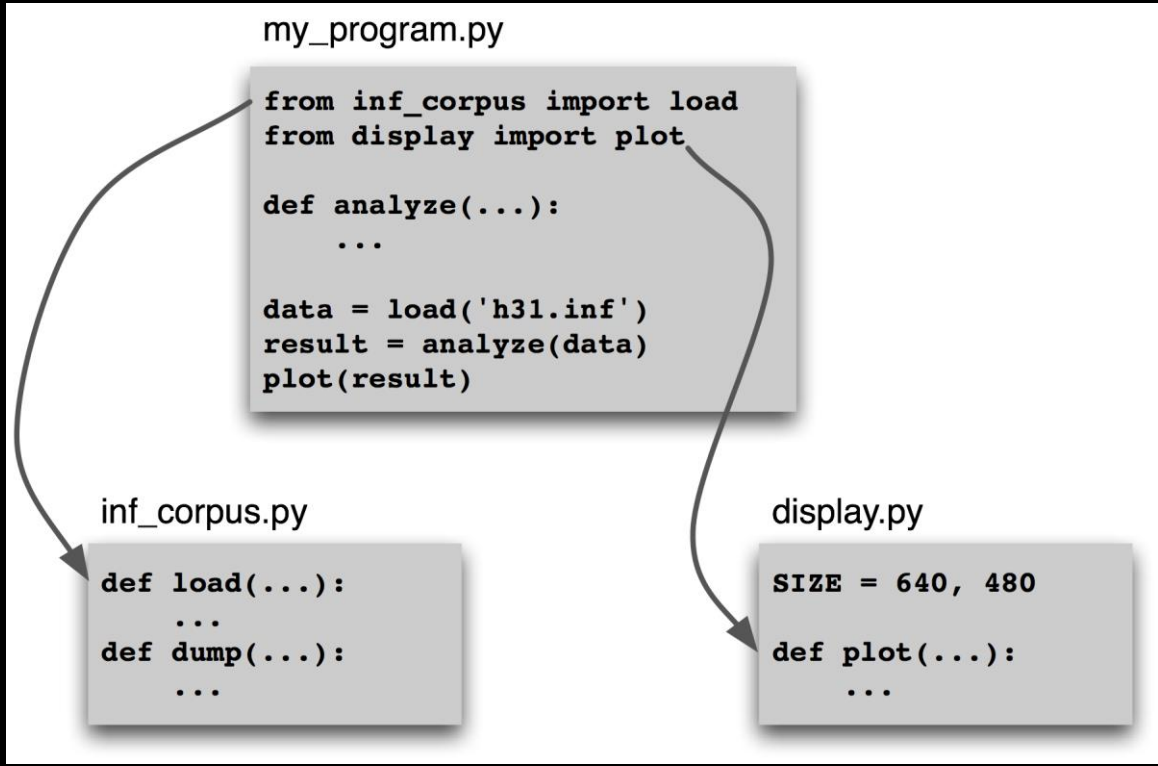
- Problems in general – Big and complex
- Hence we require lots of coding lines
- Making complex to simple – Modules / Functions
- Each module has one function like loading data, printing output or calculating some particular operation
- Main module or control module - One module which controls the overall flow
- Modules – Duplication, repetition or decisions
- Modules interconnected for problem salvations
- Main programs – calling sub functions

INTERACTIVITY CHART



INTERACTIVITY CHART




my_program.py
from inf_corpus import load
from display import plot
def analyze(...):
...
data = load('h31.inf')
result = analyze(data)
plot(result)
inf_corpus.py
def load(...):
...
def dump(...):
...
display.py
SIZE = 640, 480
def plot(...):
...

Premanand S

DEVELOPING INPUT PROCESS OUTPUT (IPO) CHART

STEP 3

DEVELOPING IPO CHART

- Extends and organizes the info from PAC
- More insight than PAC, inputs, processing and output
- IPO = PAC + IC

Input / Data	Processing	Module	Output
All input data from PAC	All processing steps involved in HIPO	Module / Function reference from IC	All output details from PAC

EXAMPLE 1 FOR IPO

- Write IPO that asks customer to enter the distance of a trip in miles, fuel consumption for the car, and the average cost for fuel. Calculate and display the fuel needed and estimated cost of the trip among the friends.

Input / Data	Processing	Module	Output
Distance in Kms Mileage of our car Cost for fuel	Distance detail Mileage details Total fuel consumption Estimated cost for trip Display total fuel & estimated cost	Distance Mileage Fuel Trip cost Total	Total fuel amount Estimated trip cost

EXAMPLE 2 IPO

- Write an IPO to compute and display the temperature inside the earth in Celsius and Fahrenheit, where Celsius = 10 * (depth) + 20 & Fahrenheit = 1.8 * (Celsius) + 32

Input / Data	Processing	Module	Output
From user	Celsius = 10 * (depth) + 20 Fahrenheit = 1.8 * (Celsius) + 32	Celsius Fahrenheit	Temperature

ASSIGNMENTS FOR IPO

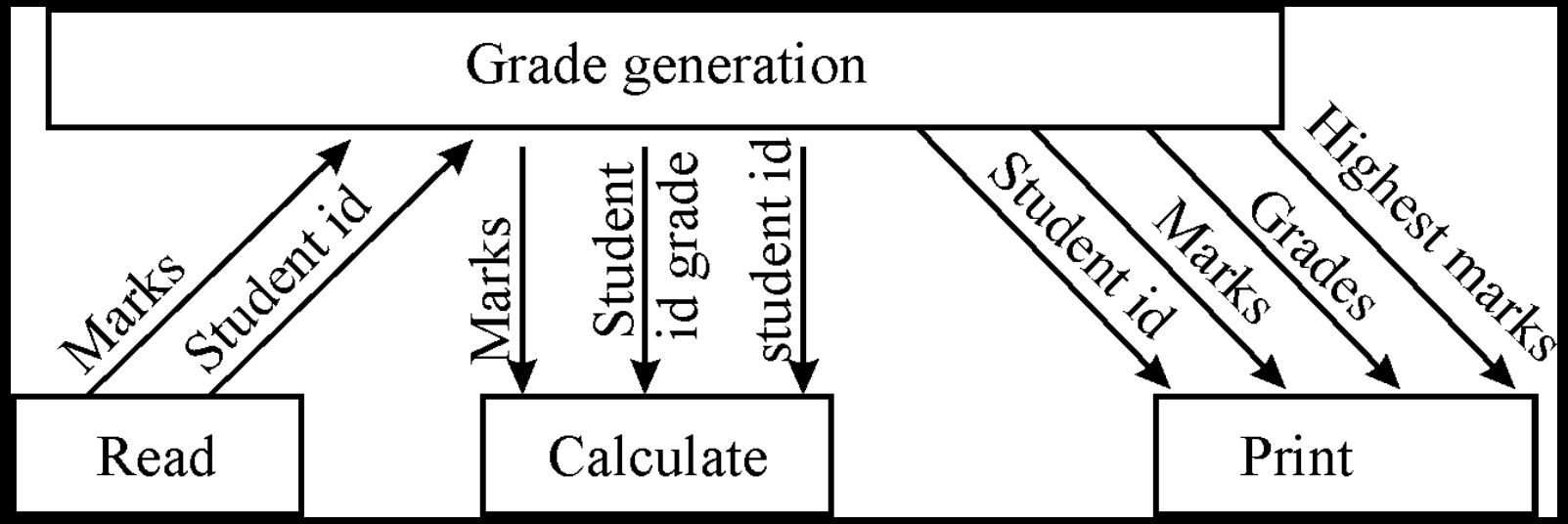
- Calculate average of five numbers by using PAC
- Convert the distance in kilometres to miles where, 1mile = 1.609km by using PAC
- Calculate the gross pay of an employee given the hours worked and rate of pay. The gross pay is calculated by multiplying the hours worked by the rate of pay.
- Analyse the problem for area of circle, area = pi*radius*radius

Premanand S

COUPLING DIAGRAM

Step 4

- The coupling diagram defines the data that gets transferred from one module to another module.



Premanand S

DATA DICTIONARY

Step 5

- A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project.
- It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation.

DATA

emlployee_id	first_name	last_name	nin	dept_id
44	Simon	Martinez	HH 45 09 73 D	1
45	Thomas	Goldstein	SA 75 35 42 B	2
46	Eugene	Cornelsen	NE 22 63 82	2
47	Andrew	Petculescu	XY 29 87 61 A	1
48	Ruth	Stadick	MA 12 89 36 A	15
49	Barry	Scardelis	AT 20 73 18	2
50	Sidney	Hunter	HW 12 94 21 C	6
51	Jeffrey	Evans	LX 13 26 39 B	6
52	Doris	Berndt	YA 49 88 11 A	3
53	Diane	Eaton	BE 08 74 68 A	1

DATA DICTIONARY (METADATA)

Column	Data Type	Description
emlployee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current postion title, e.g. Secretary
dept_id	int	Employee deparmtnet. Ref: Departmetns
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date.

Premanand S

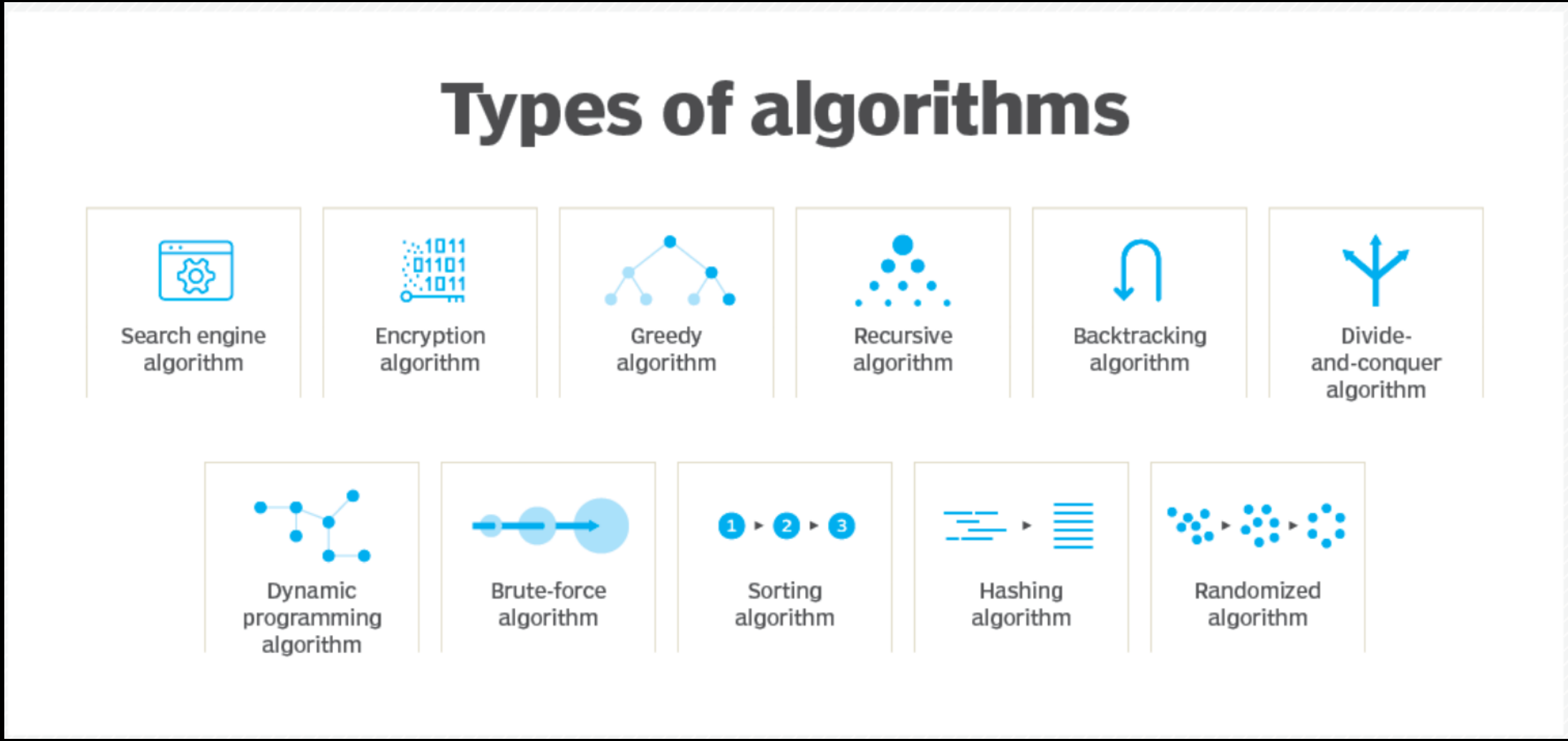
ALGORITHMS

STEP 6

ALGORITHM

- Effective method – finite list of well defined instructions for any function from start position to end, including all inputs and process.
- Used for any kind of problems
- Step before writing any programming code in English language for easy understanding
- Step by step to solve any problem

ALGORITHMS - TYPES



STRUCTURING THE PROGRAM

- Use of functions or modules
- Use of logic structures like,
  - Sequential structure – one after another
  - Decision structure – branches for decision making
  - Loop structure – executing many times
  - Case structure – executing one set of instruction out of many
- Improve readability by using logic structures, variables, indentation and proper documents.

SEQUENCE LOGIC STRUCTURE

- All instructions are executed one after another
- Sample,

Step1 : Start

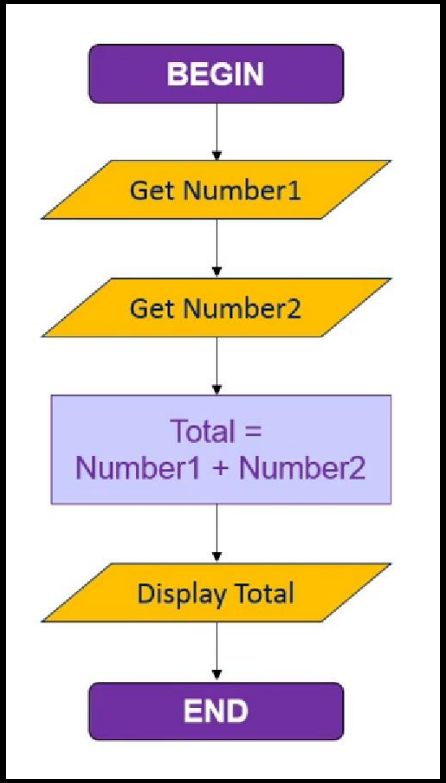
Step2 : Get the 1st number, Number1

Step3 : Get the 2nd number, Number2

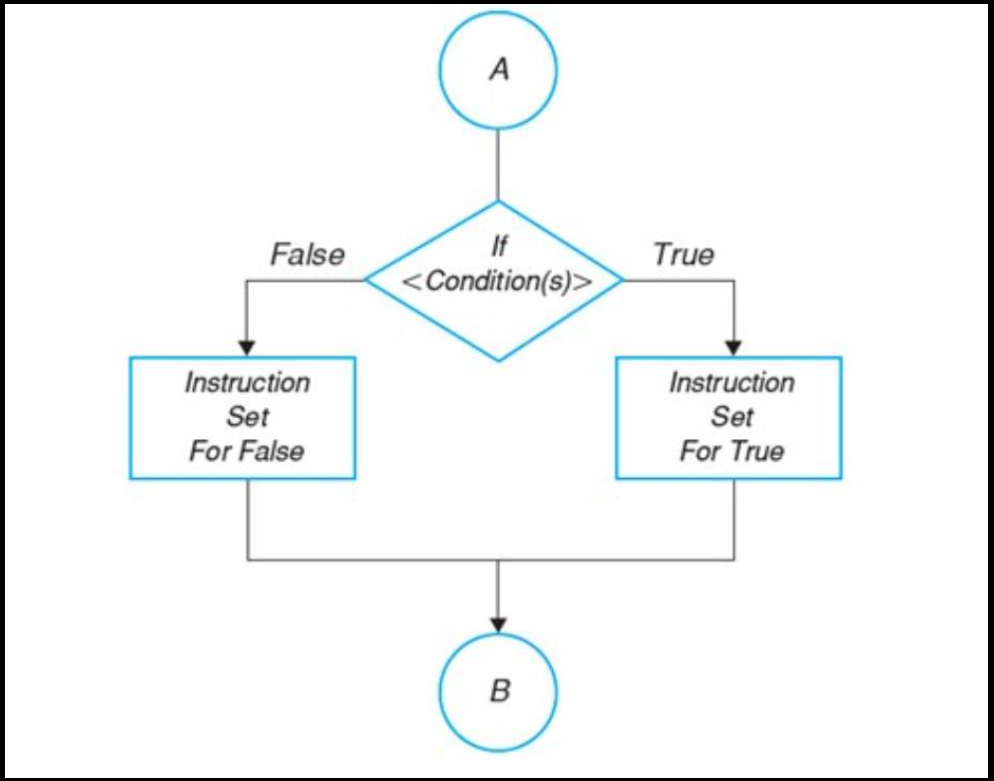
Step4: Total = Number1 + Number 2

Step5 : Display the number

Step6 : Stop



DECISION LOGIC STRUCTURE / SELECTION LOGIC



- Some portion of the instruction is executed based on condition (Yes/No, True/False and …)
- If condition is True / Yes, then this part will execute it first or else, other condition i.e., False/No
- Sample,

Step1 : Start

Step2 : Getting numbers

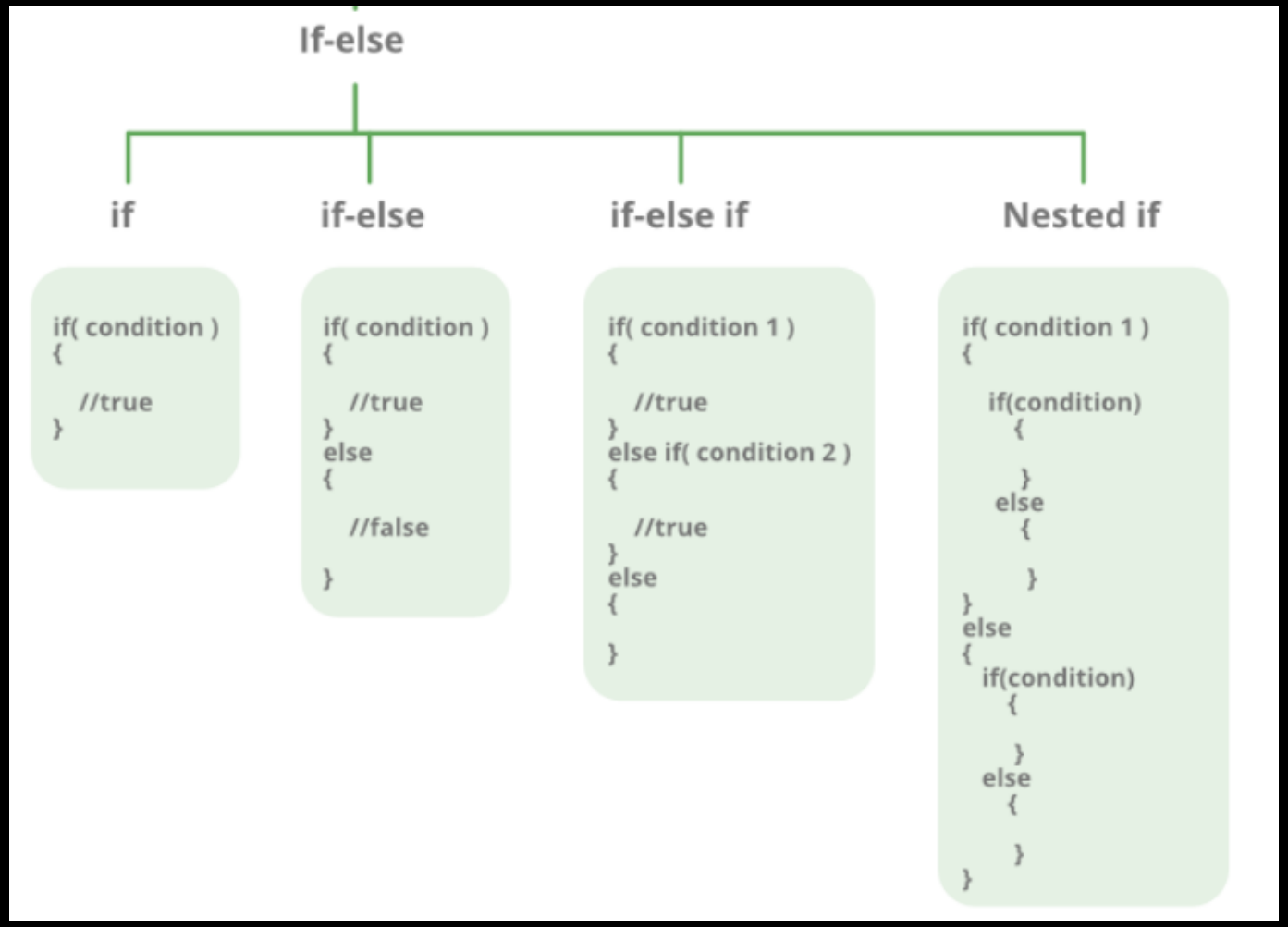
Step3 : Passing to the condition, if its true, it will move on step 4 or else step 2.

Step4: It will pass to the final step for display

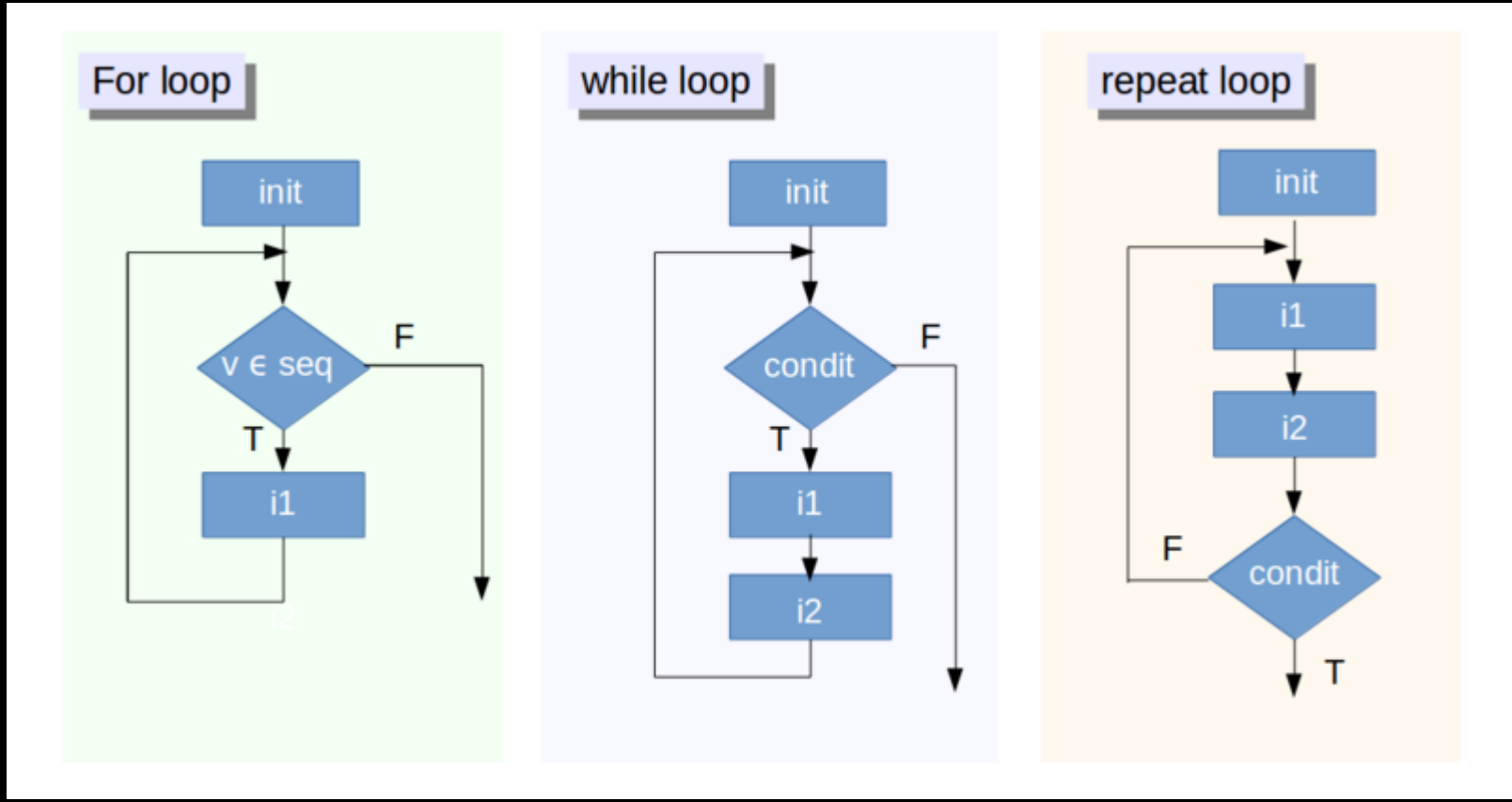
Step5 : In step3, if the condition is False/No, it will pass to step2 and again process, till it will satisfy the condition.

Step6 : Stop

DIFFERENT DECISION LOGIC STRUCTURES



LOOP LOGIC STRUCTURES



FOR LOOP

Python for loop

A for loop is **used for iterating over a sequence and iterables** (like range, list, a tuple, a dictionary, a set, or a string).

```
for i in range(5):
    statement 1
    statement 2
    ...
    statement n
else:
    statement(s)
```

Definite iterations.
(Total 5 iterations)

Indentation
Loop body is must be properly indented

Body of for loop
Execute till the last item of a sequence

Else Block (optional)
Execute only when for loop executes normally

NESTED FOR LOOP

Nested For loop

```
for i in range(1, 11):
    for j in range(1, 11):
        print(i*j, end=" ")

    print('')
```

Outer Loop

Inner loop

Body of inner loop

Body of Outer loop

WHILE LOOP

Python While loop

While loops **repeat the same code as long as a certain condition is true**

```
while Condition:
        statement 1
        statement 2
        …
        statement n
else:
        statement(s)
```

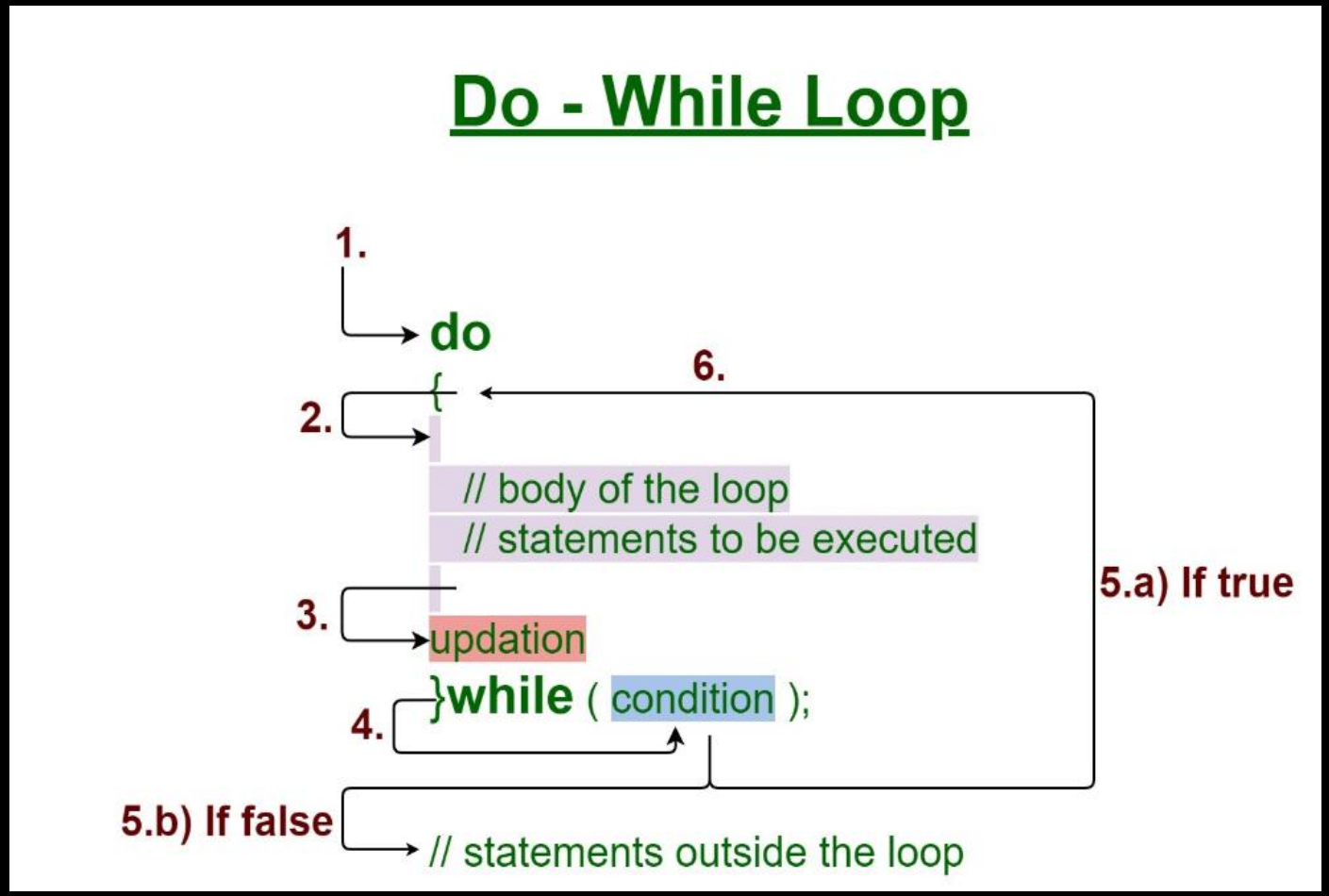
Expression that returns True or False

Indentation
Loop body is must be properly indented

Body of while loop
Execute as long as condition is True

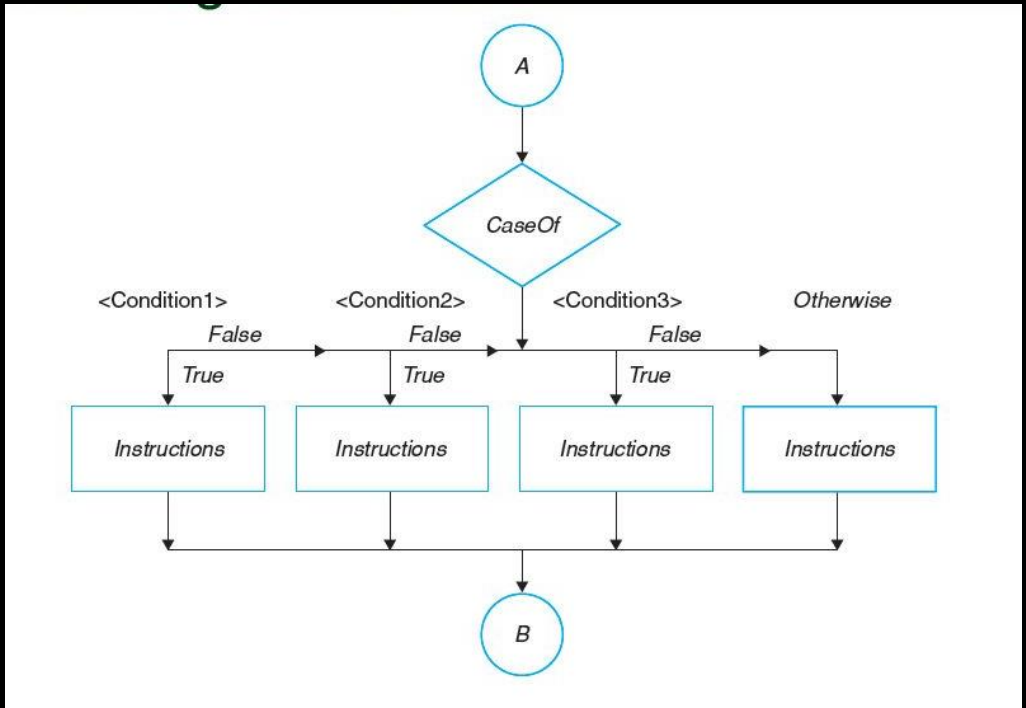
Else Block (optional)
Execute only when while loop executes normally

DO WHILE


Do - While Loop
1.
do
{
2.
// body of the loop
// statements to be executed
3.
updation
}while (condition);
4.
6.
5.a) If true
5.b) If false
// statements outside the loop

WHILE LOOP VS DO WHILE LOOP

CASE LOGICAL STRUCTURE



ALGORITHM 1: ADD TWO NUMBERS ENTERED BY THE USER

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read values num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.

sum←num1+num2

Step 5: Display sum

Step 6: Stop

ASSIGNMENTS

- **Find the largest number among three numbers**
- **Find Roots of a Quadratic Equation $ax^2 + bx + c = 0$**
- **Find the factorial of a number**
- **Check whether a number is prime or not**
- **Find the Fibonacci series till the term less than 1000**

ALGORITHMS - REPRESENTATION

- Algorithms can be represented in many forms, and among that there are few methods which will be used globally.
- There are few methods to represent algorithms in other forms,
  - Flowchart
  - Pseudocode
- It shows the logic behind algorithms without implementation.
- Non programmer also understand the flow.

Premanand S

FLOWCHART

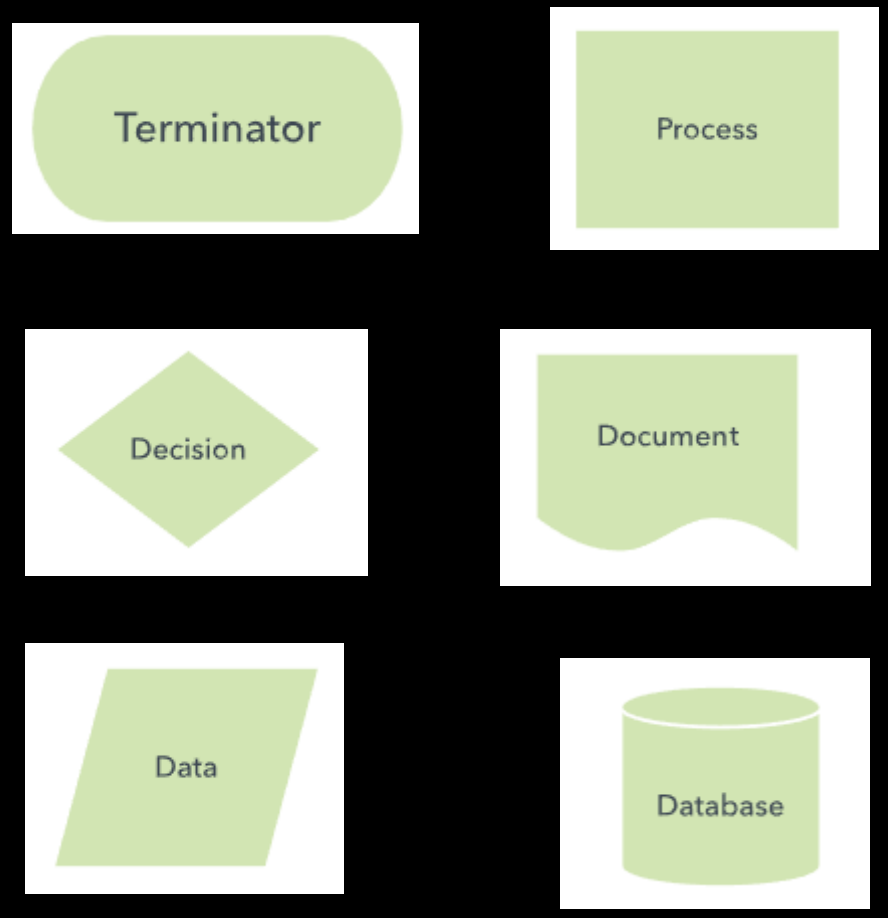
STEP 7

FLOWCHART

- Graphical representation – individual steps
- Logical design of a program
- Actual program can be developed from this
- Flowchart ~ Blue print of a building
- Independent of any programming language
- Standard norms

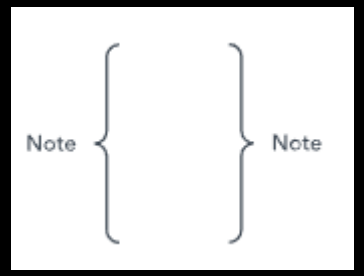
STANDARD SYMBOLS USED IN FLOWCHART

- Terminal / Terminator
- Process
- Decision
- Document
- Data ~ input / output
- Stored data

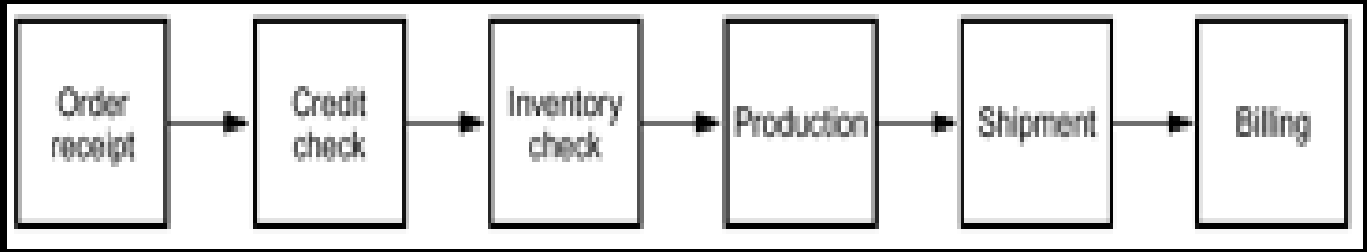


STANDARD SYMBOLS USED IN FLOWCHART

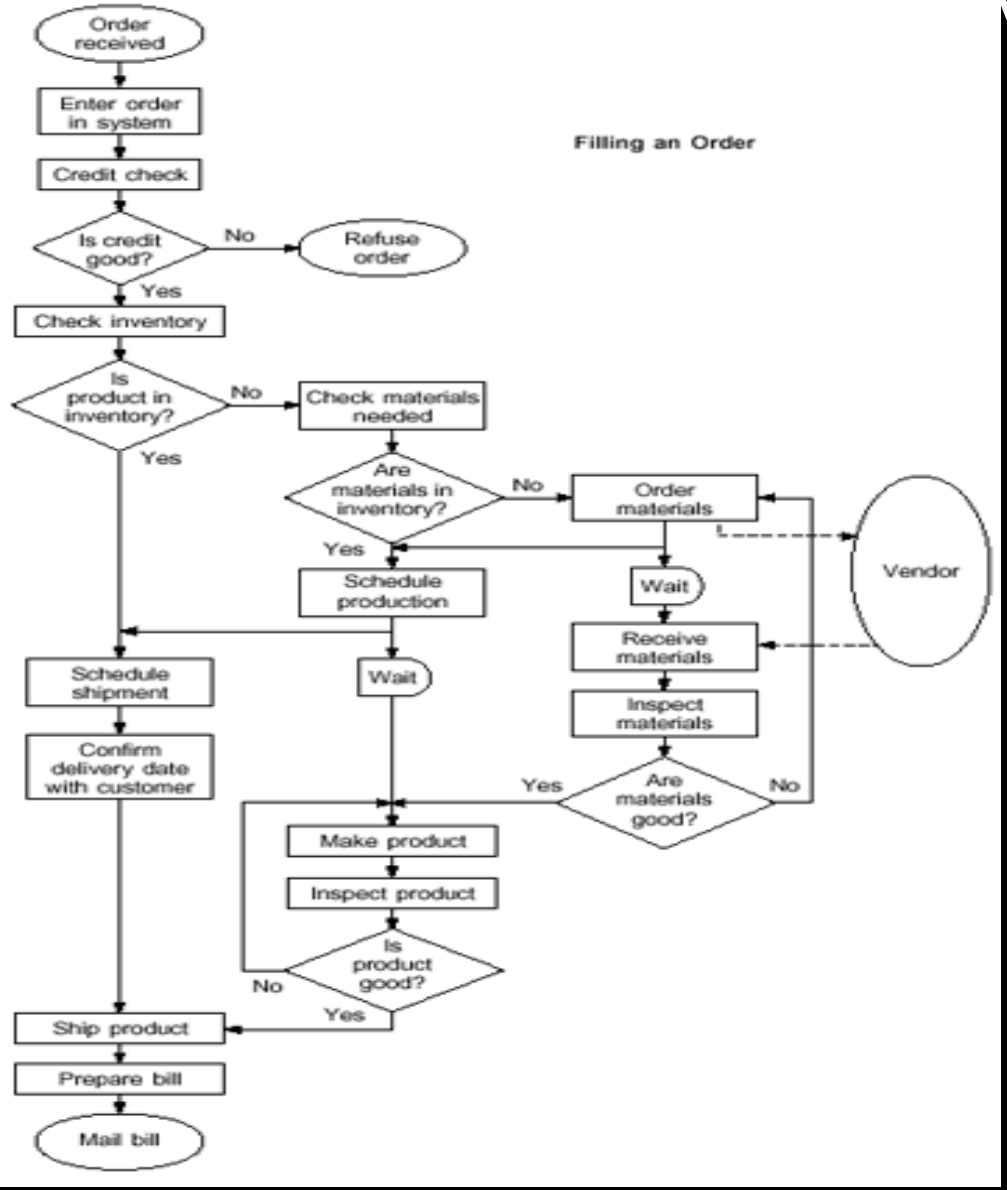
- Flow arrow
- Comment & Annotation
- Predefined process
- On-page connector
- Off-page connector

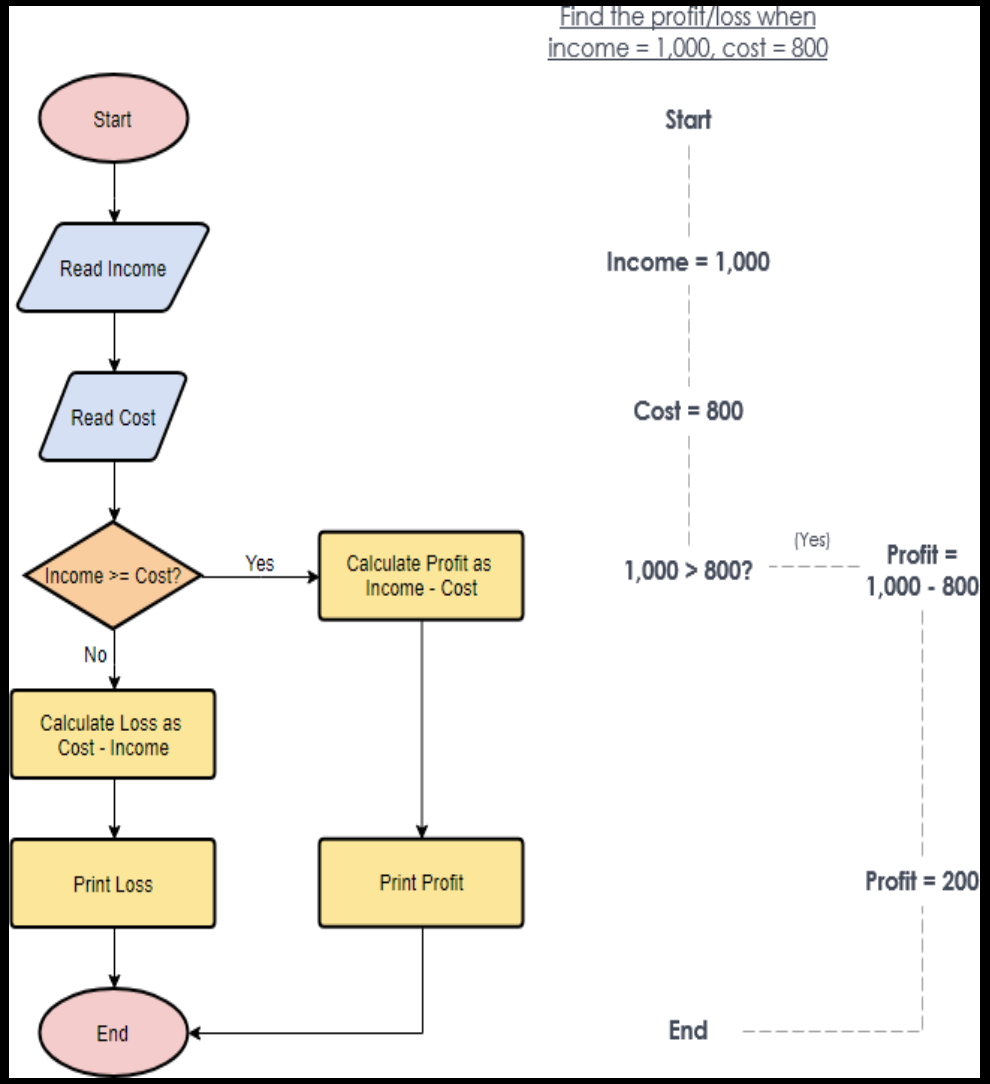


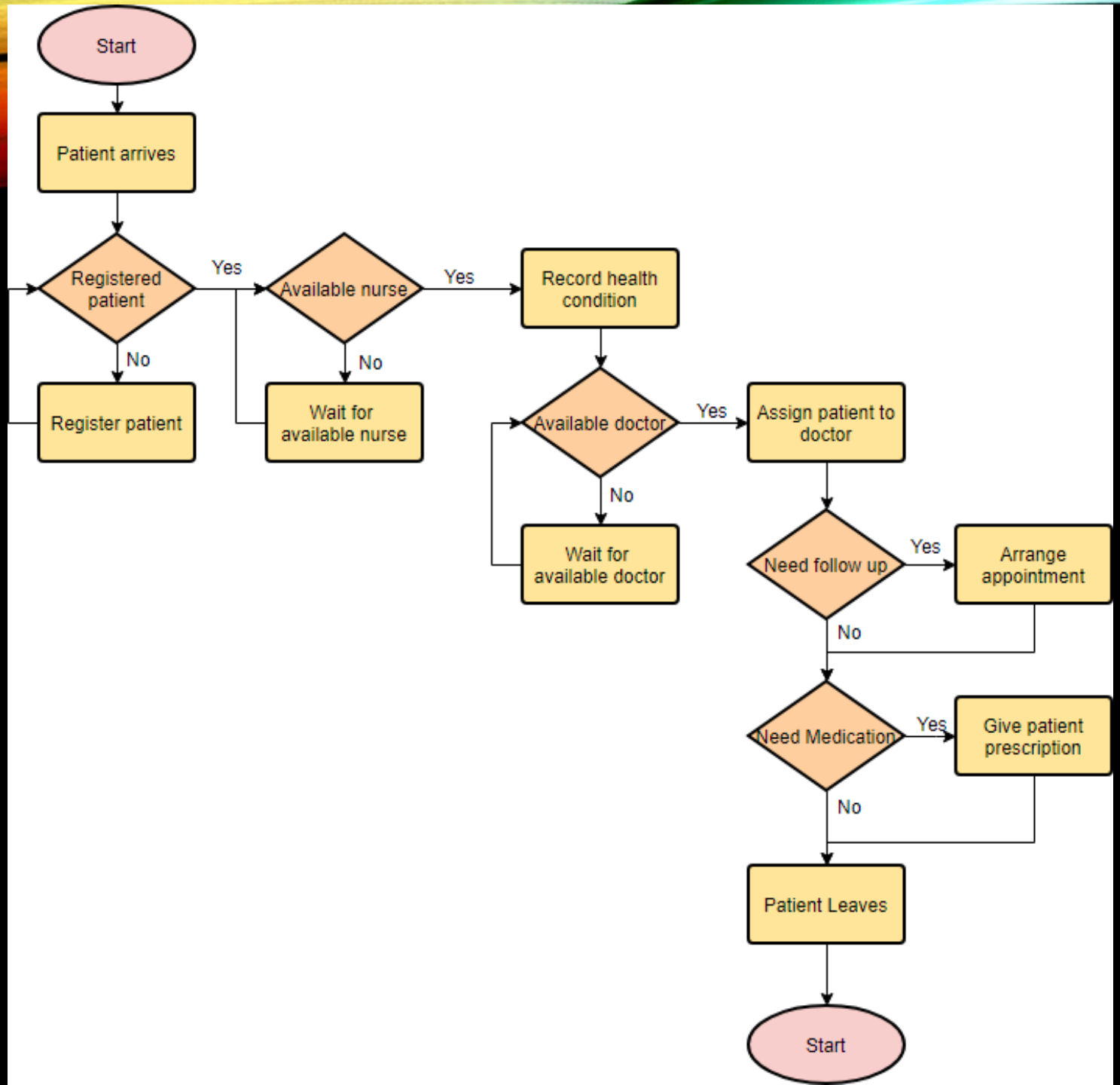
HIGH LEVEL FLOWCHART



DETAILED FLOWCHART

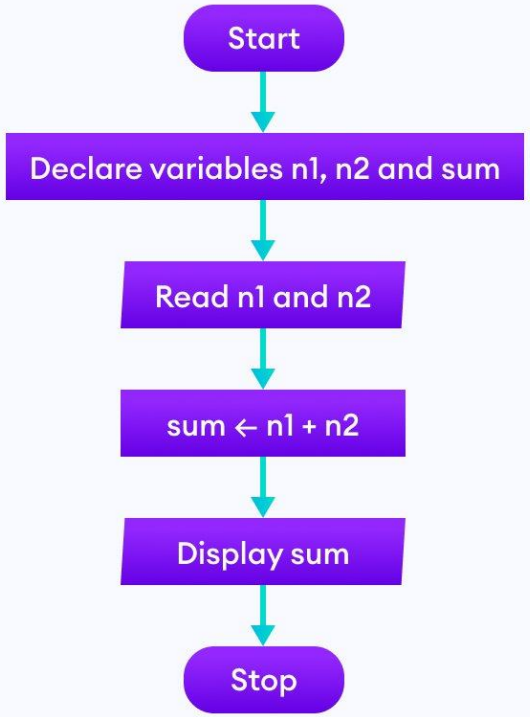



Find the profit/loss when
income = 1,000, cost = 800
Start
Read Income
Read Cost
Income >= Cost?
Yes
No
Calculate Profit as
Income - Cost
Calculate Loss as
Cost - Income
Print Loss
Print Profit
End
Start
Income = 1,000
Cost = 800
1,000 > 800?
(Yes)
Profit =
1,000 - 800
Profit = 200
End

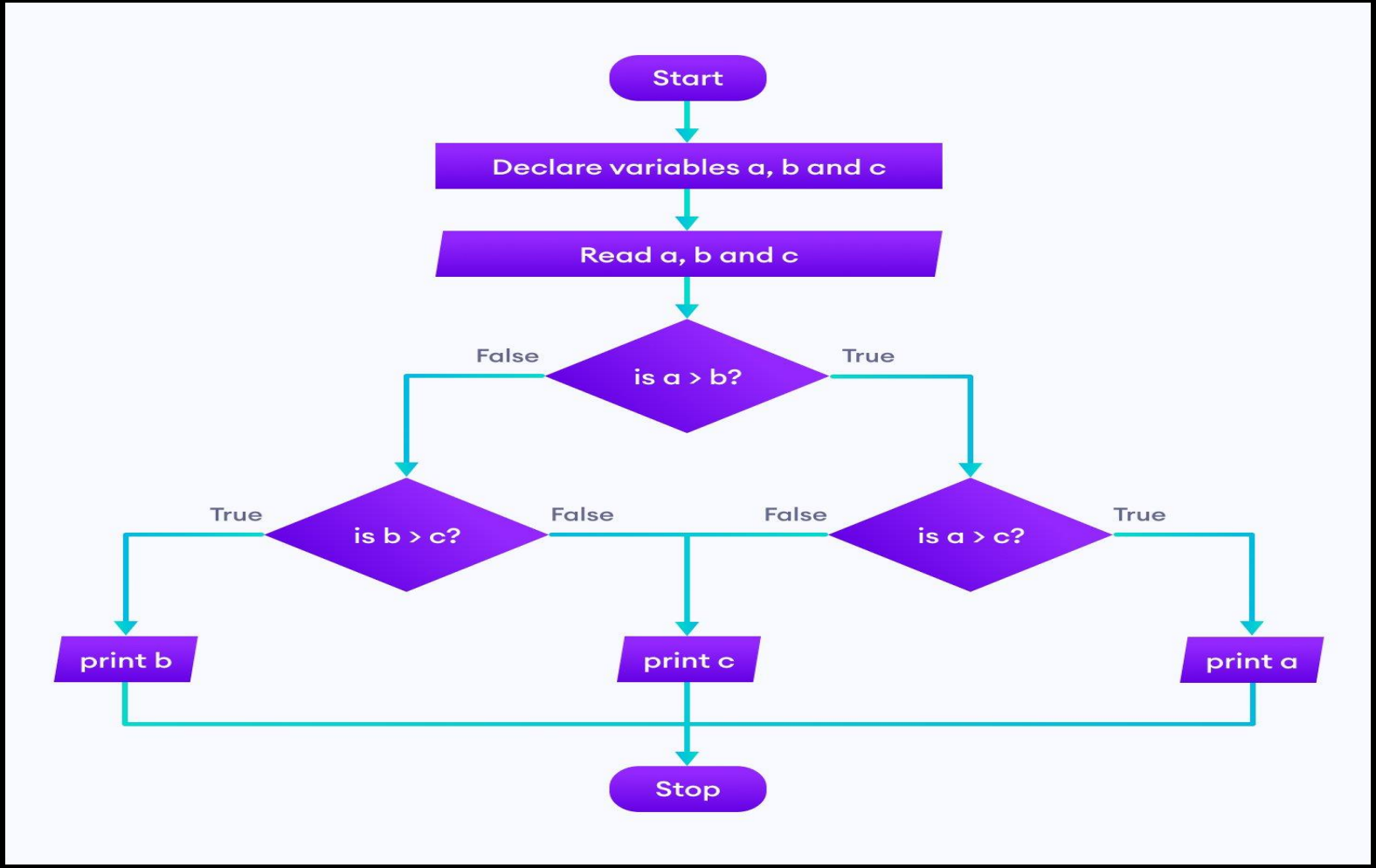

Start
Patient arrives
Registered patient
Yes
No
Register patient
Available nurse
Yes
No
Wait for available nurse
Record health condition
Available doctor
Yes
No
Wait for available doctor
Assign patient to doctor
Need follow up
Yes
No
Arrange appointment
Need Medication
Yes
No
Give patient prescription
Patient Leaves
Start

Premanand S

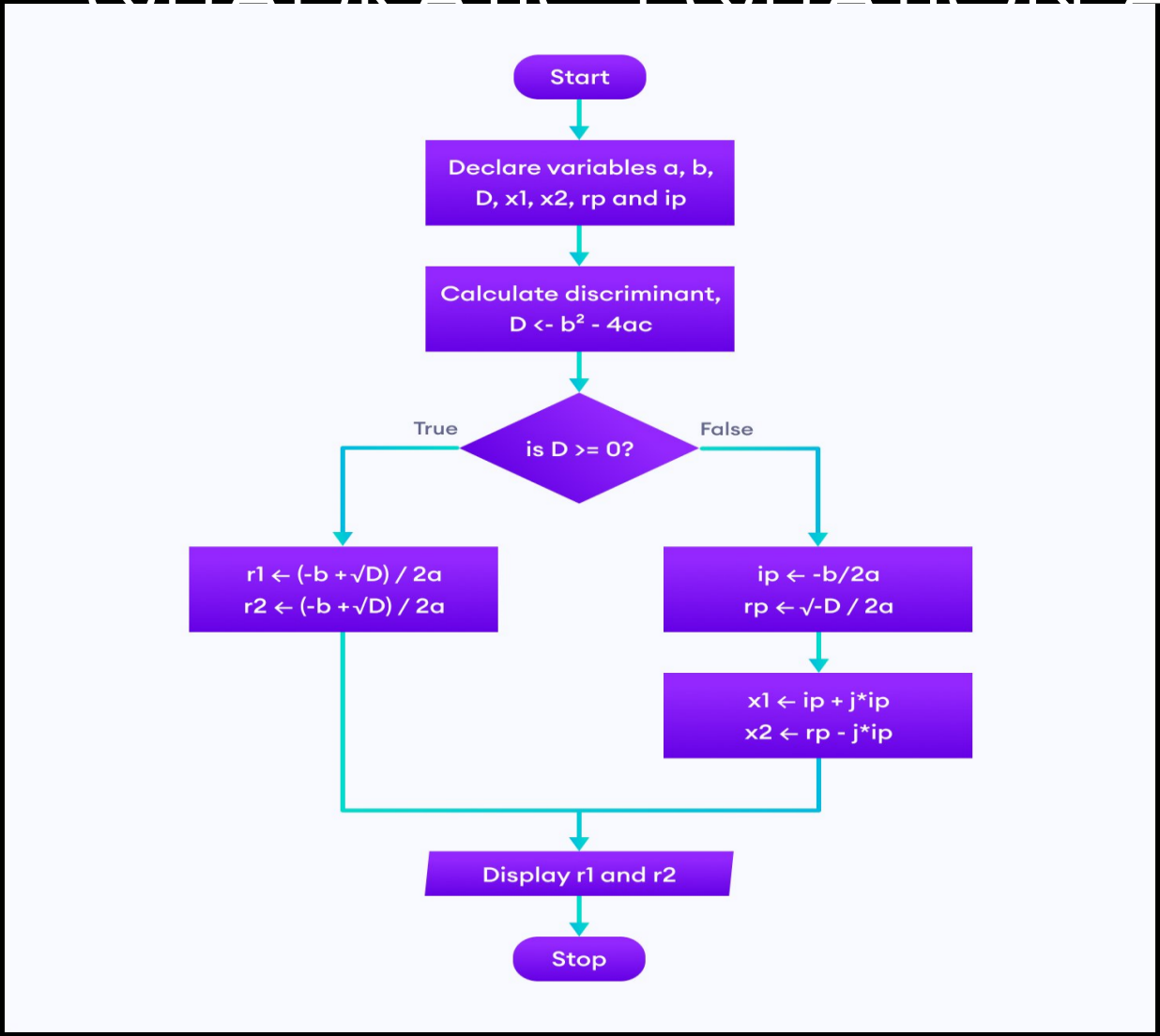
EXAMPLE 1 FLOWCHART FOR ADDING TWO NUMBERS FROM USER



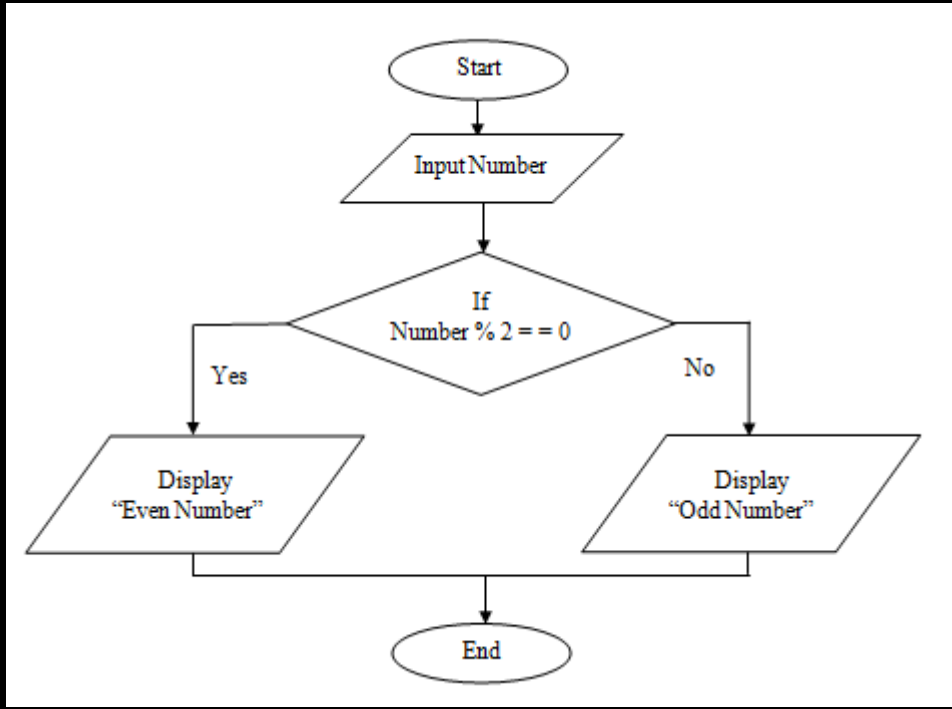
EXAMPLE 2: FIND THE LARGEST AMONG THREE DIFFERENT NUMBERS ENTERED BY THE USER.



EXAMPLE 4: FIND ALL THE ROOTS OF A QUADRATIC EQUATION $AX^2+BX+C=0$



EXAMPLE 5 FLOWCHART FOR ODD OR EVEN NUMBERS



ASSIGNMENTS FOR FLOWCHART

- Read the sequence of numbers, find the average of the number and print the average.
- **Find the Fibonacci series till term≤1000.**
- **Hiring process in any company starting from advertisement till offer letter**
- **Draw a flowchart for - Should I Break Up With Him**

Premanand S

PSEUDOCODE

STEP 8

PSEUDOCODE

- Pseudo – False / Fake; Code – Computer / programming code
- Artificial / informal language – Programmers
- Text based tool
- In place of symbols or flowchart to describe the logic of a program
- **You make the rules yourself**

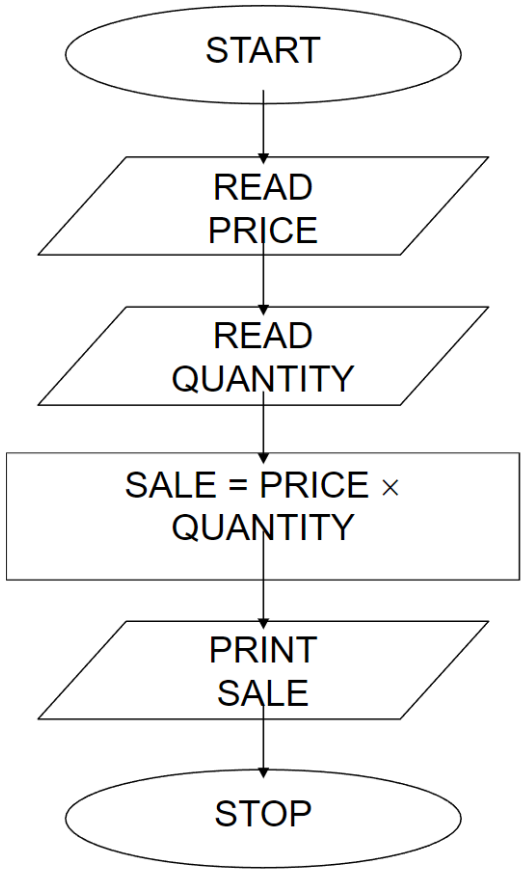
- Note: Statements with ‘Dependency’ are to be intendent like while, do, if, switch…

KEYWORDS FOR PSEUDOCODE

- Input: **READ, OBTAIN, GET**
- Output: **PRINT, DISPLAY, SHOW**
- Compute: **COMPUTE, CALCULATE, DETERMINE**
- Initialize: **SET, INIT**
- Add one: **INCREMENT, BUMP, DECREMENT**

- **Do not include data declaration**

EXAMPLE 1 WRITE THE PSEUDOCODE FOR READ THE PRICE AND QUANTITY OF THE PRODUCT AND DISPLAY THE SALE DETAILS



Algorithm:

Start

Read price, quantity
Sale = price x quantity
Print Sale

End

EXAMPLE 2 AREA OF RECTANGLE

- READ height of Rectangle
- READ width of Rectangle
- COMPUTE area as height times width
- PRINT area of rectangle

EXAMPLE 3 COMPARISON

```
If student's grade is greater than or equal to 60
        Print "passed"
else
        Print "failed"
```

DON'T

```
// program to find the largest of 2 numbers

IF n1>n2
      Print "n1"
ELSE
      Print "n2"
```

```
IF number 1 is greater than number 2
      Print "number 1 is greater"
ELSE
      Print "number 2 is greater"
```

ASSIGNMENTS

- Write Pseudocode for adding Two Numbers
- Write pseudocode for calculating Area (l*h) and Perimeter (2*(l+h)) of Rectangle
- Write pseudocode for issue of driver license (if else)
- Write pseudocode for a given number is positive or negative
- Write pseudocode to read 50 numbers and find their sum and average. (for loop)

Premanand S

REVISION

PROBLEMS SOLVING - OVERALL

Mr. Jones always gives True/False tests to his class. His tests always have 20 questions. The maximum class size is 35. He needs a program that will calculate the students' grades based on the best score.

Grade

A will range from the best score, to the best score minus 2.

B will range from the best score minus 3, to the best score minus 4.

C will range from the best score minus 5, to the best score minus 6.

D will range from the best score minus 7, to the best score minus 8.

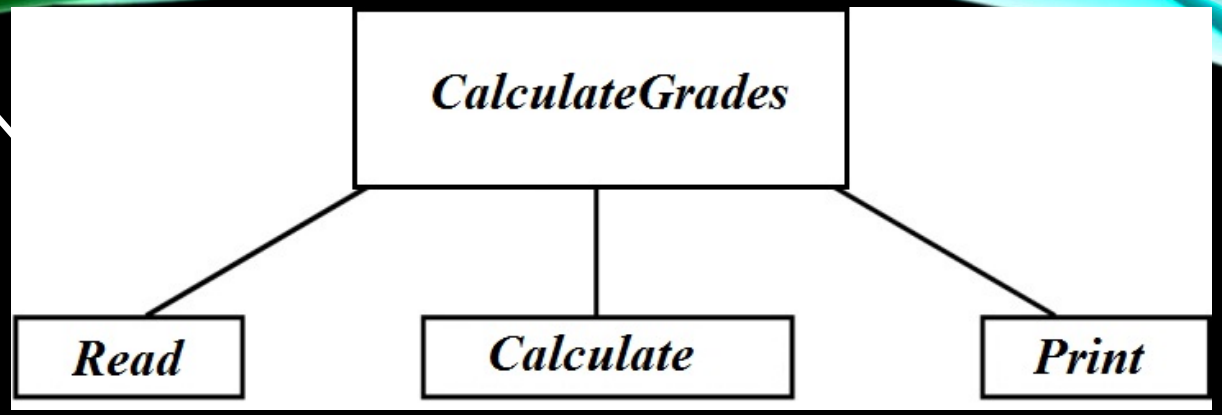
F will be anything below the best score minus 8.

Each student's ID and test answers will be entered. The output will be each student's ID, number correct, and grade, along with the single highest score for the class. Develop a solution for Mr. Jones's problem. Use four one-dimensional arrays—one for the correct scores and the other three for the needed output.

STEP 1: PROBLEM ANALYSIS CHART (PAC)

	Required Result
Class strength = 35 Total questions = 20 A if the marks is greater than the best M minus 2 B if the marks is greater than the best M minus 3 to best M minus 4 C if the marks is greater than the best M minus 5 to best M minus 6 D if the marks is greater than the best M minus 7 to best M minus 8 F if the marks is below the best M minus 8	Calculating the student's grades based on the best (marks) M. Taking the student's id, marks received. It finds out the (highest) H marks. Then generate the grades based on the H marks.
Processing	Solution Alternative
Generation of the grades based on the H marks obtained along with the student id and the marks for each student.	Generate the solution for a particular student instead of showing the result for the entire class.

STEP 2: IN



- The interactivity chart separates the solution into modular parts.
- Working of different modules,

  Calculate Grades module controls the solution.

  Read module reads data like student id, grades and the marks obtained.

  Calculate module calculates the H marks obtained and the grades based on the H marks of the class.

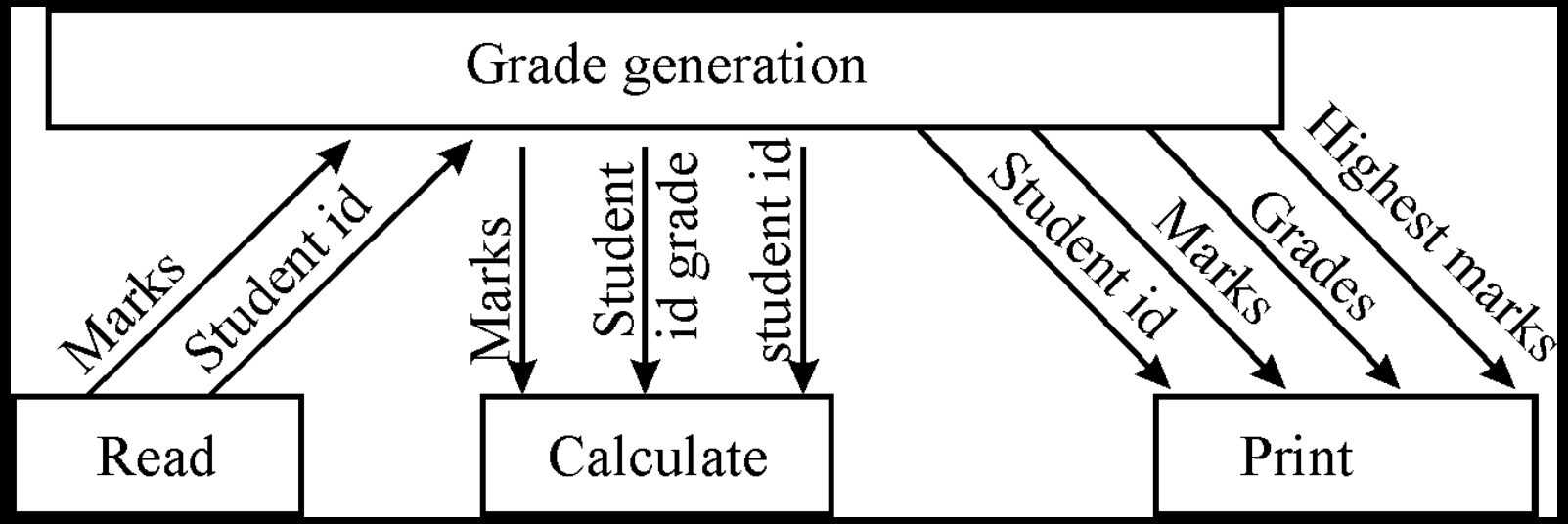
  Print module prints the student id, grades, marks obtained and the H marks obtained.

STEP 3: INPUT PROCESS OUTPUT (IPO) CHART

Processing	Module Reference	Output
1. Read data like student id and marks obtained.	READ	Please enter the student id and the marks for 35 students.
2. Calculate the H marks and the grades of the students.	CALCULATE	
3. Print the result	PRINT	The marks, student id, grades and the H marks of all the students.

STEP 4: COUPLING DIAGRAM

- The coupling diagram defines the data that gets transferred from one module to another module.



STEP 5: THE DATA DICTIONARY

Item	Variable name	Data Type	Module (s)	Scope
Student id	*sid*	integer	Read	LOCAL
marks	*M*	integer	Read	LOCAL
grades	*g*	character	Read	LOCAL
H Marks	*H*	integer	Read	LOCAL

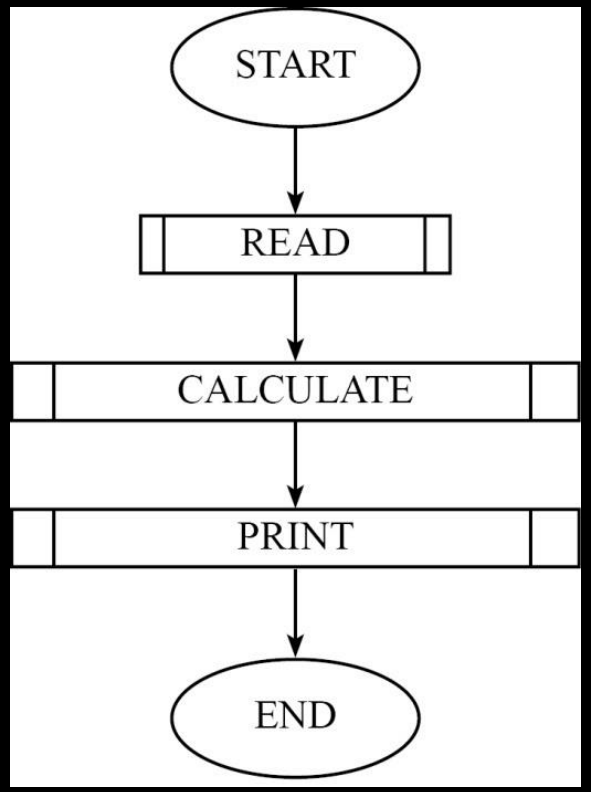
STEP 6: ALGORITHM

- The algorithm defines the order of steps that are initiated to solve a particular problem.

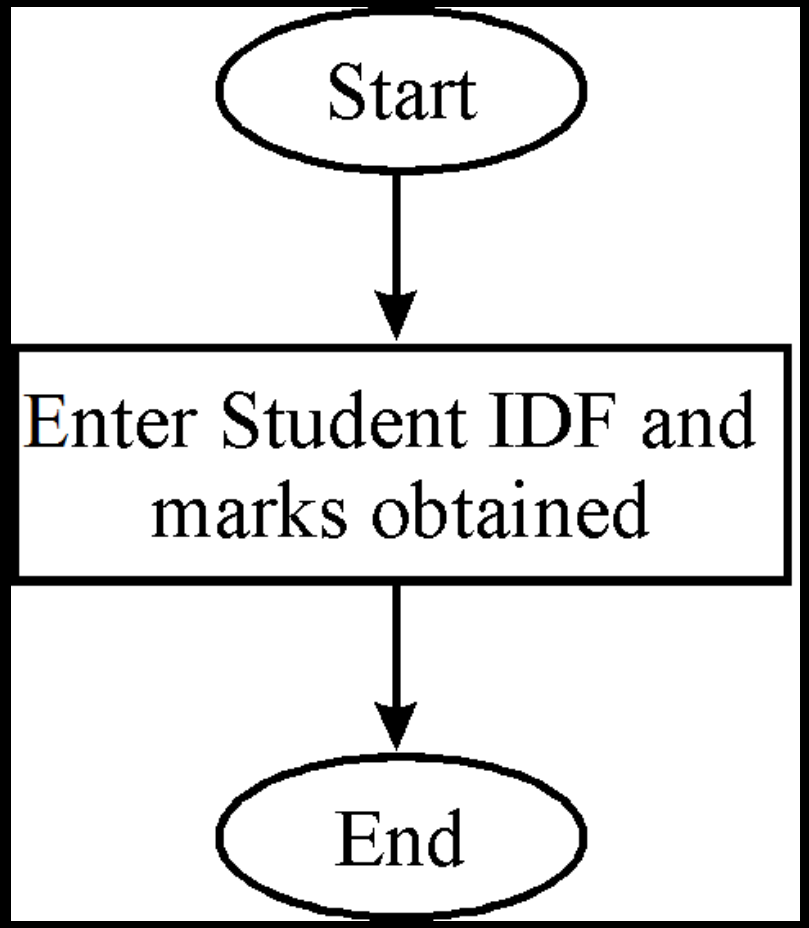
(1) Start.
(2) Read the data like the marks and student id.
(3) Calculate the H marks obtained.
(4) Calculate the grades obtained based on the given set of rules.
(5) Display the student id, marks, grades and the H marks

STEP 7: FLOWCHART

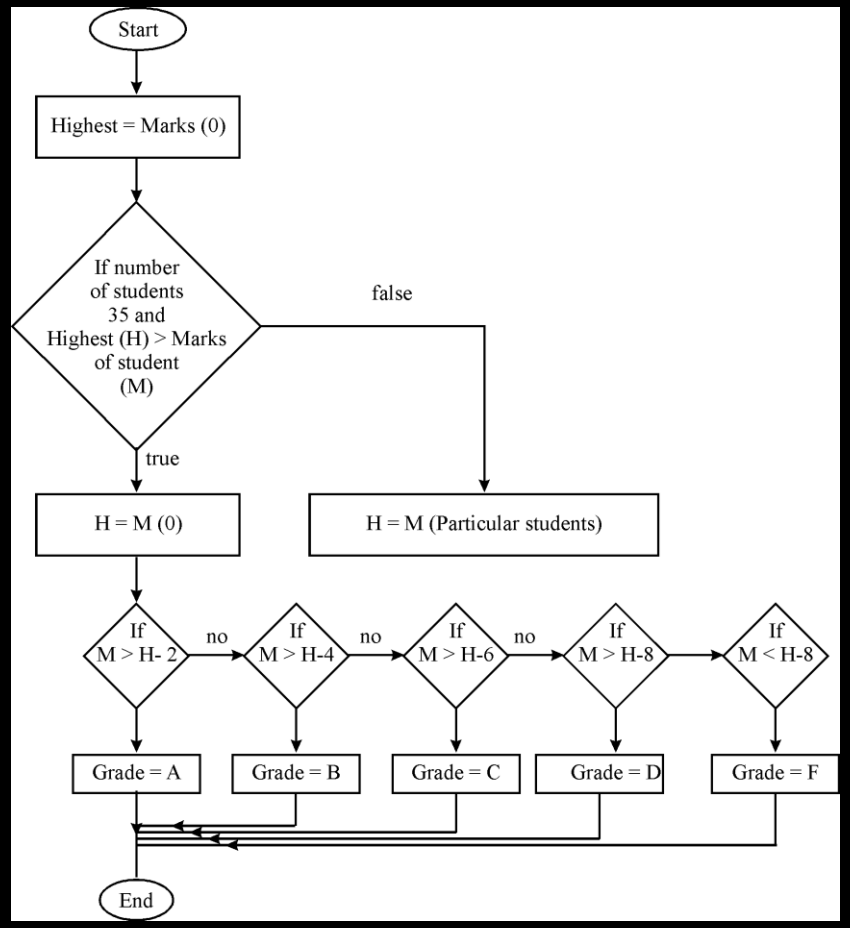
- The flowcharts are a graphic interpretation of the algorithm.



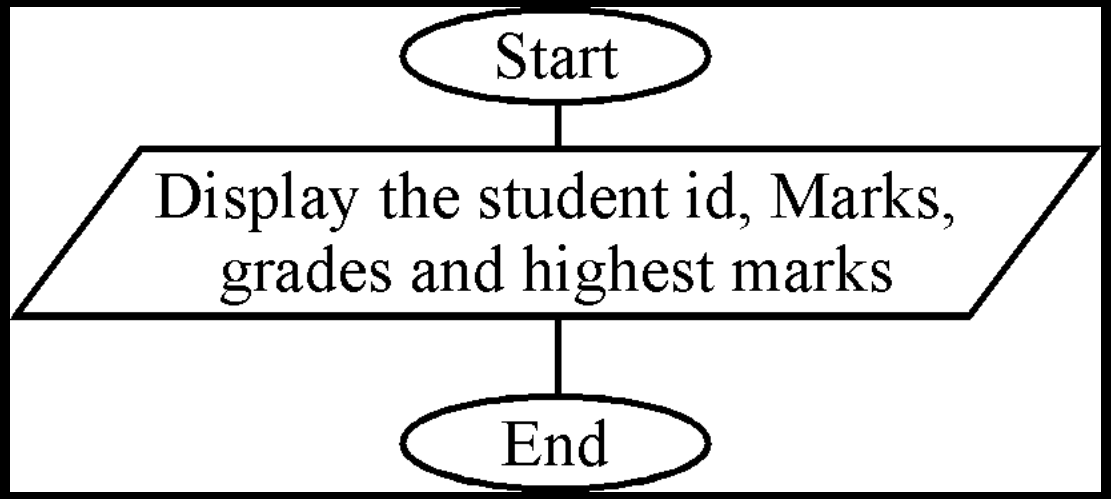
STEP 7: FLOWCHART (CONT…) – READ MODULE



STEP 7: FLOWCHART (CONT…) – CALCULATE MODULE



STEP 7: FLOWCHART (CONT…) – PRINT MODULE



STEP 8: PSEUDOCODE

```
sid[35],Grades[35],M[35],High[35];
for( i=0;i< 35; i++)
{
print("Enter the student id and M for student number % d");
scanf("%d %c", &sid[i], &Ms[i]);
}
for ( i=0; i<35; i++)
{
if (M[i]>M[i+1])
H = M[i];
}
```

```
if ( M[i] >= (H-2))
grade[i] = 'A';
if( M[i] >= (H-4) && M[i] <=(H-3))
grade[i] = 'B';
if( M[i] >= (H-6) && M[i] <=(H-5) )
grade[i] = 'C';
if( M[i] >= (H-8) && M[i] <=(H-7) )
grade[i] = 'D';
if( M[i] <=(H-8) )
grade[i] = 'F';
for(i=1 ;i < 35 ; i ++)
print ID[i], M[i], grade[i], H;
```

ASSIGNMENT 1

A restaurant manager wants to know how many employees are needed at the restaurant each hour of the day. The minimum number of employees needed at any hour is 3. After that, one additional employee is required for each 20 customers. The restaurant is open 24 hours a day. The manager has counted the number of customers each hour for 14 days. The manager will use the average number of customers for each hour over the 14 days to calculate the needed number of employees for each hour. Develop a solution to output the needed number of employees per hour. (There is no such thing as a partial employee.)

ASSIGNMENT 2

- An instructor has 30 students in her class. Each student is identified by a number from 1 to 30. Grades are stored in a one-dimensional array. The instructor would like to enter a student number and have the student's test score printed on the monitor. Develop a solution to output the needed information.

“

COMPUTER SCIENCE IS A SCIENCE OF ABSTRACTION CREATING THE RIGHT MODEL FOR A PROBLEM AND DEVISING THE APPROPRIATE MECHANIZABLE TECHNIQUE TO SOLVE IT.

”

- ALFRED AHO & JEFFERY ULLMAN