PROBLEM SOLVING AND PYTHON PROGRAMMING

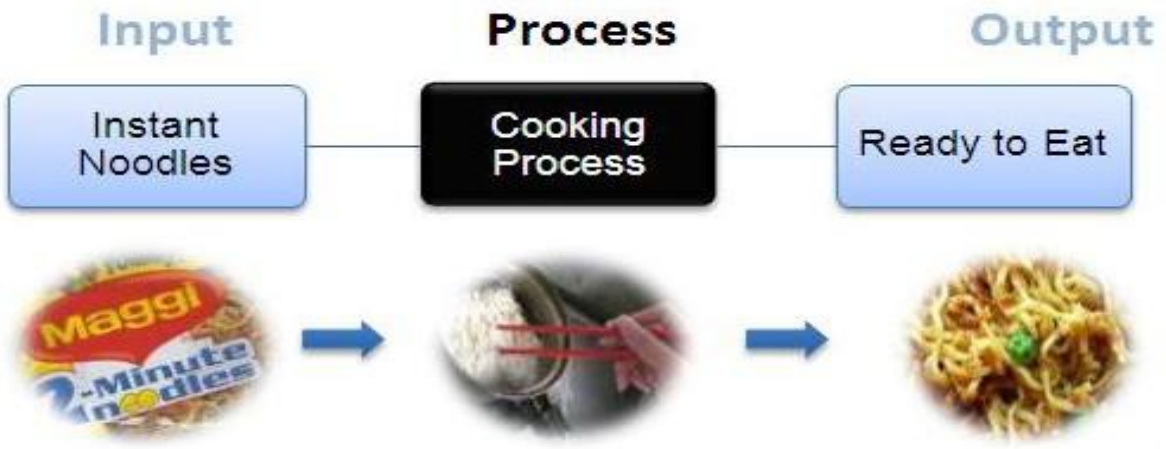
Subject code: GE3151

APEC

Dr. C. Dhaya, Professor & Head

Define Computer

- A computer is an electronic data processing device, which accepts data input, processes the data input, and generates the output in a required format.



Define Computer

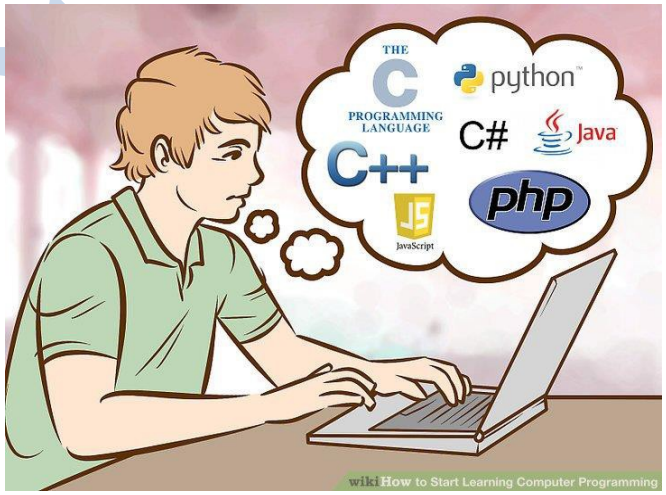
- A computer is an **electronic device** that helps people perform different tasks.
- Computers allows us to
  - Write letters
  - Create pictures
  - Make calculations
  - Play games
  - Send electronic mail (e-mail) via computers.



APEC

What is a Computer Program?

- **Computer Program:**
  - The step-by-step instructions that tells a computer what to do and produces a desired result



- **Computer Programmer:**
  - A person who writes(generates) a computer program

APEC

Define Algorithm.

- The sequence of steps to be performed by the computer, to solve a problem is known as an algorithm`.

APEC

Contd...



Input



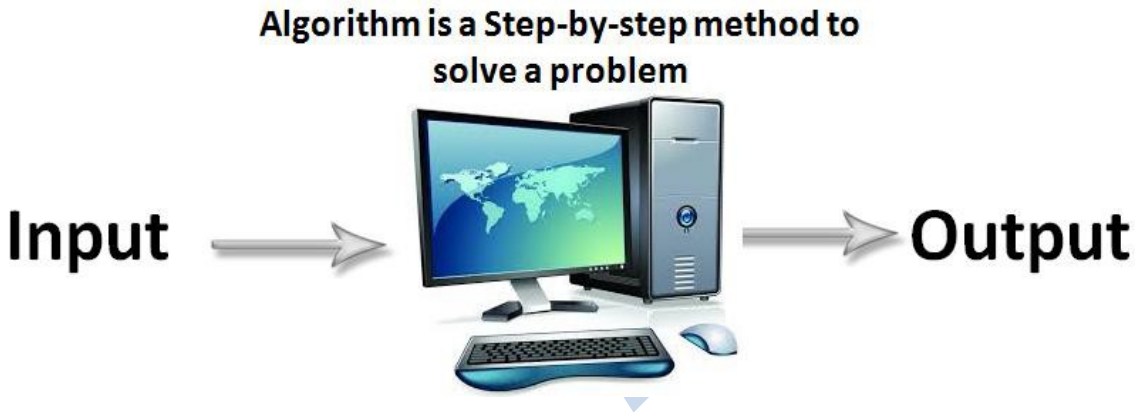


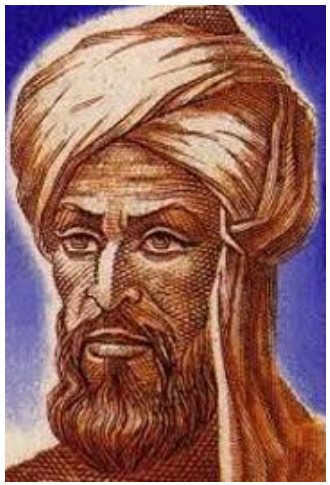
Output

ALGORITHM

Algorithm definition Contd…

- The sequence of steps to be performed by the computer, to solve a problem is known as an algorithm.





- Developed by Persian scientist, astronomer and mathematician Abdullah Muhammad bin Musa al-Khwarizmi in 9th century.
  - “The father of Algebra”

Program Vs. algorithm

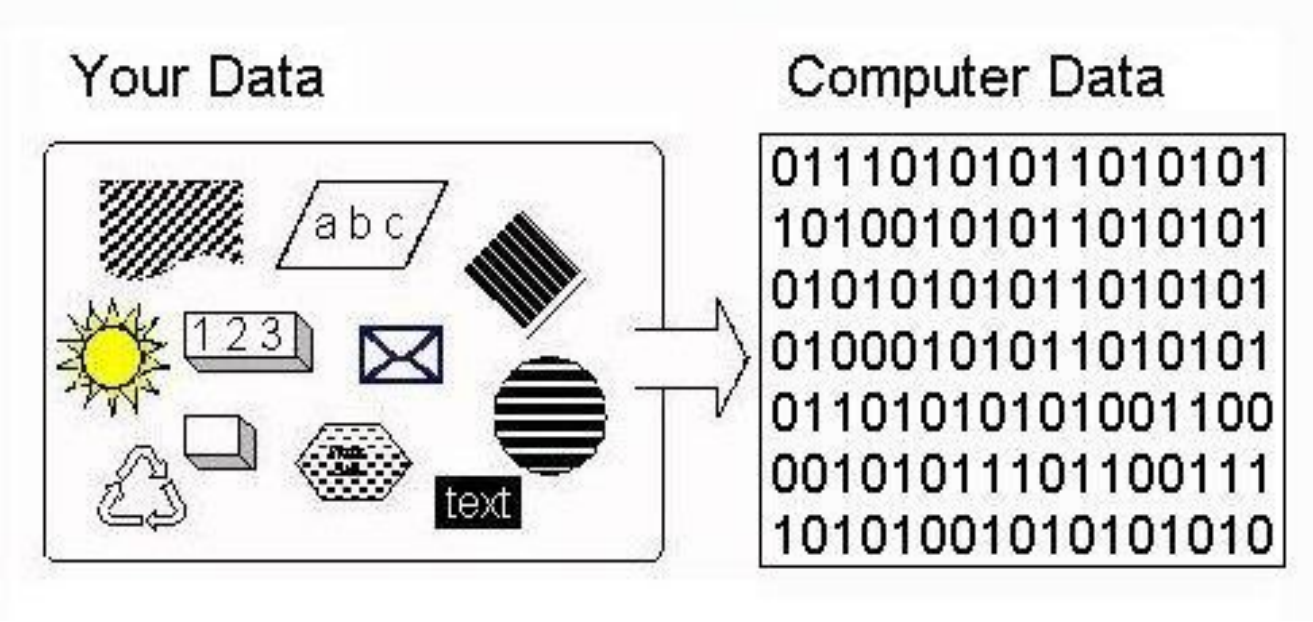
- **Algorithm:** the step-by-step procedure for solving the problem.

- **Program:** the implementation in “Computer codes” that instructs the computer how to execute the algorithm.

APEC

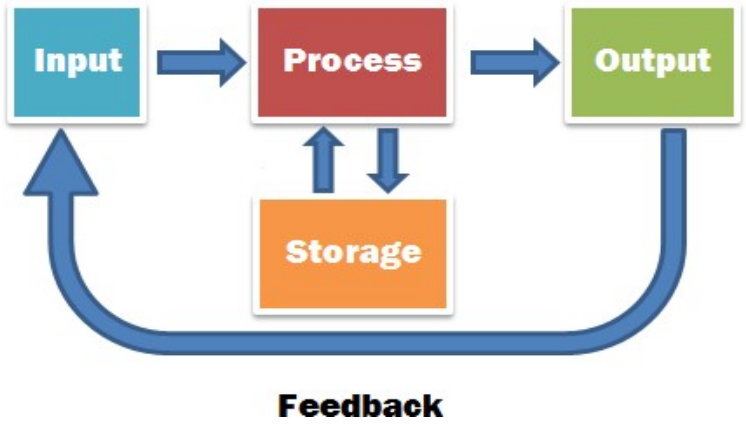
Computer Instructions

- The set of instructions that tells the computer, what to do and how to do.



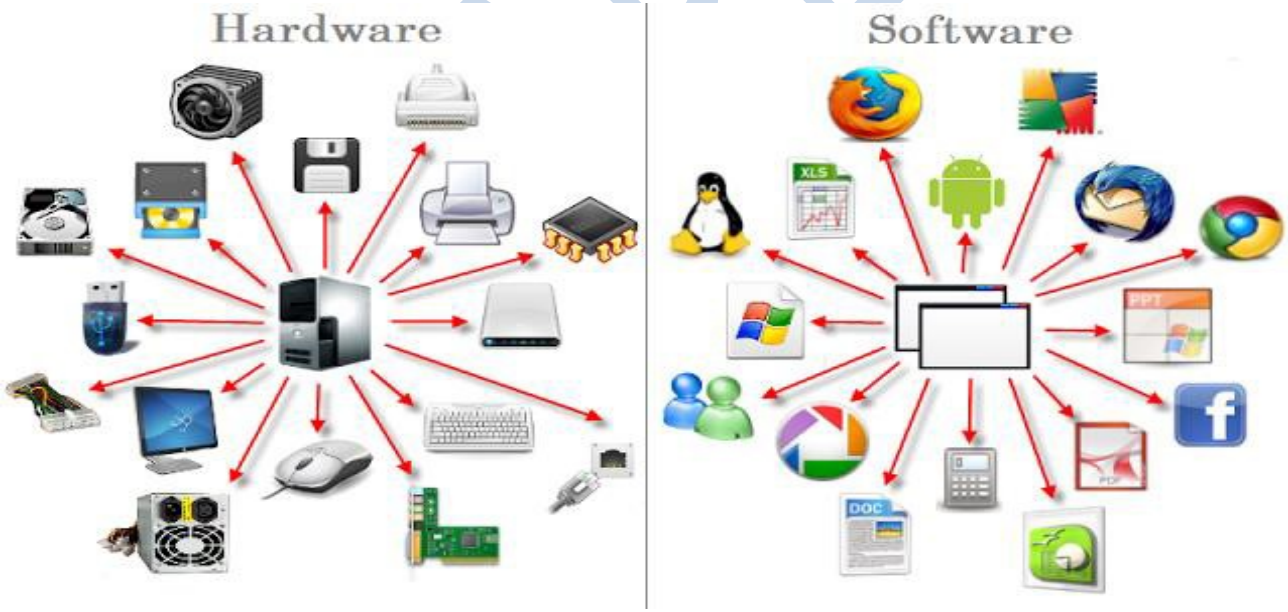
Steps involved in processing the Data

- Step 1: Takes data as input.
- Step 2: Stores the data/instructions in its memory and uses them as required.
- Step 3: Processes the data and converts it into useful information.
- Step 4: Generates the output.



Hardware and Software

- **Hardware** ➔ physical and tangible components of a computer, i.e. the components that can be seen and touched.
- **Software** ➔ Set of instructions for a computer to perform specific operations.

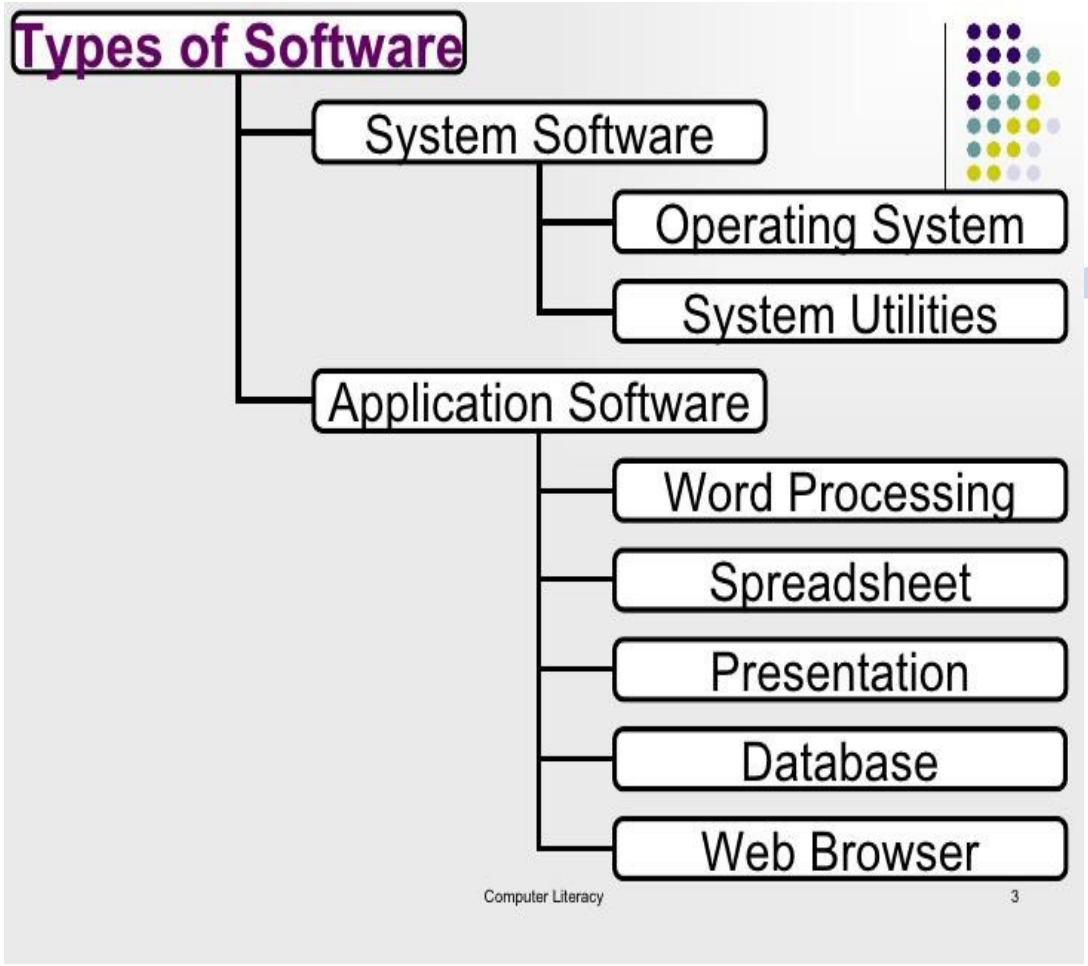


Examples of Hardware

- Input devices:
  - keyboard, mouse, etc.
- Output devices:
  - printer, monitor, etc.
- Secondary storage devices:
  - Hard disk, CD, DVD, etc.
- Internal components:
  - CPU, motherboard, RAM, etc.

APEC

Examples of Software





Application Software



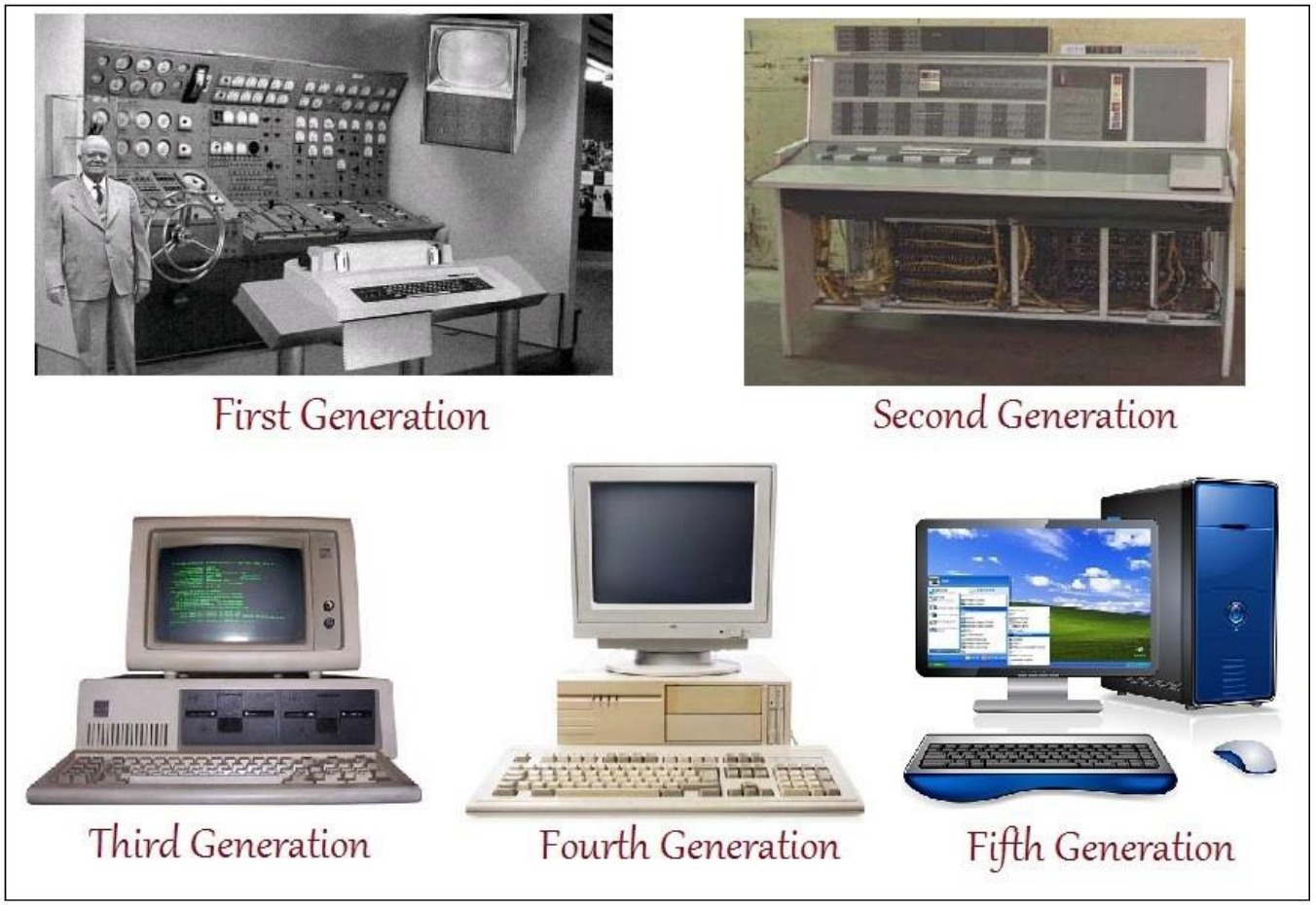
System Software

Generation Of Computers

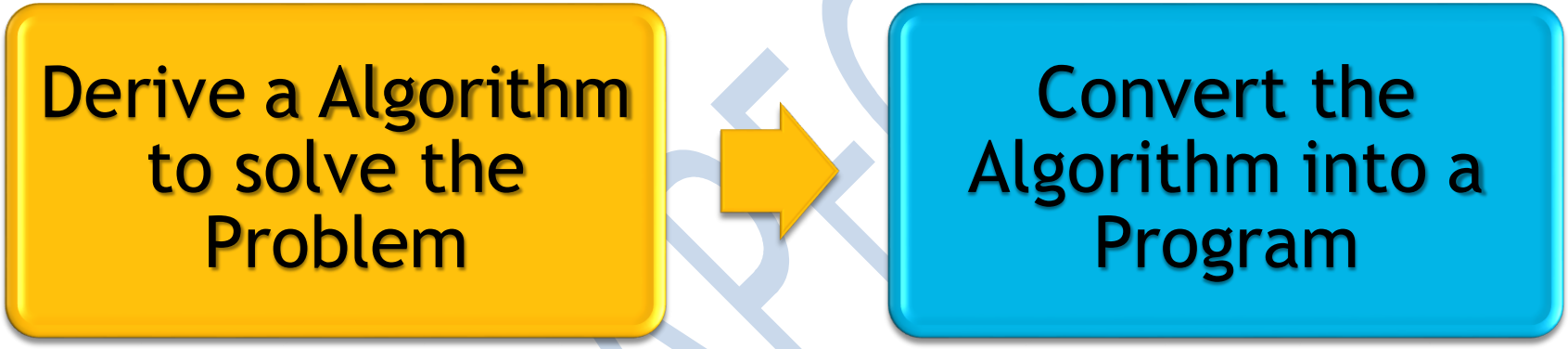
- First Generation (1940 - 1956) - Vacuum Tubes:
  - The computers of this generation were made of vacuum tubes for circuitry and magnetic drums for memory
- Second Generation (1956 - 1963) - Transistors:
  - The computers of this generation made up of transistors replacing vacuum tubes
- Third Generation (1965 - 1971) - Integrated Circuits:
  - The computers of these generations were made up of IC (Integrated Circuits)
- Fourth Generation (1971) - Microprocessors:
  - The computers of this generation saw the advent of Large Scale Integration (LSI) and Very Large Scale Integration (VLSI)
- Fifth Generation (1980 - till date) - ULSI (Ultra Large Scale Integration):
  - The computers of this generation use massively parallel computing/processing.

APC

Generation Of Computers



Phases of Algorithmic problem solving



Example

Problem Statement (or) Problem:

Write a Program to add two numbers

Algorithm

Step 1: Start
Step 2: Enter two numbers in a and b
Step 3: Add a and b and store in c
Step 4: Display c
Step 5: End

APEC

Contd…

Program

addition of nubmers.c

```c
#include<stdio.h>
int main()
{
    double a, b , result =0 ;
    printf("Enter first number :  ");
    scanf("%lf",&a);

    printf("Enter Second Number :  ");
    scanf("%lf",&b);

    result = a + b;
    printf("Addition of two numbers is :  %lf", result);

    return 0;
}
```

Program ➔ “Implementation of the Algorithm” in any of the Programming languages

In Computer Science



Input

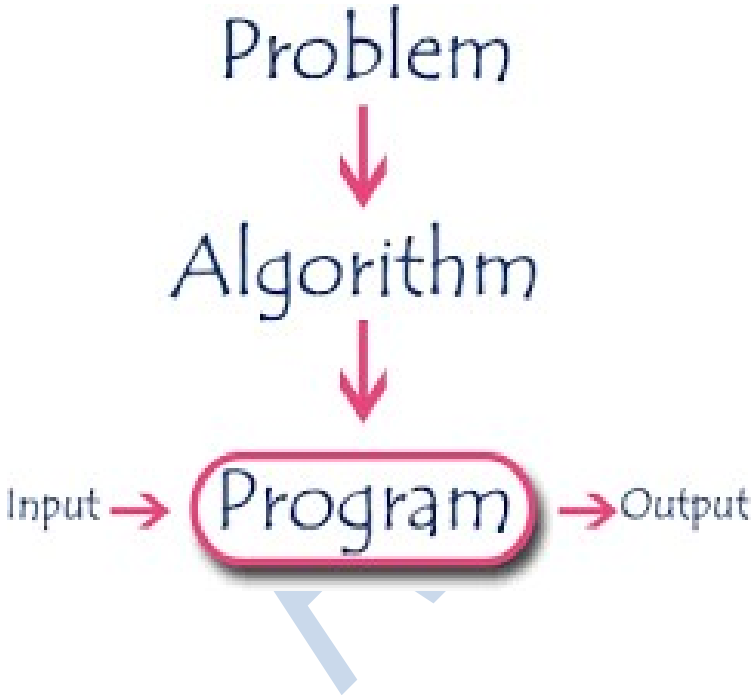


Processing or Steps

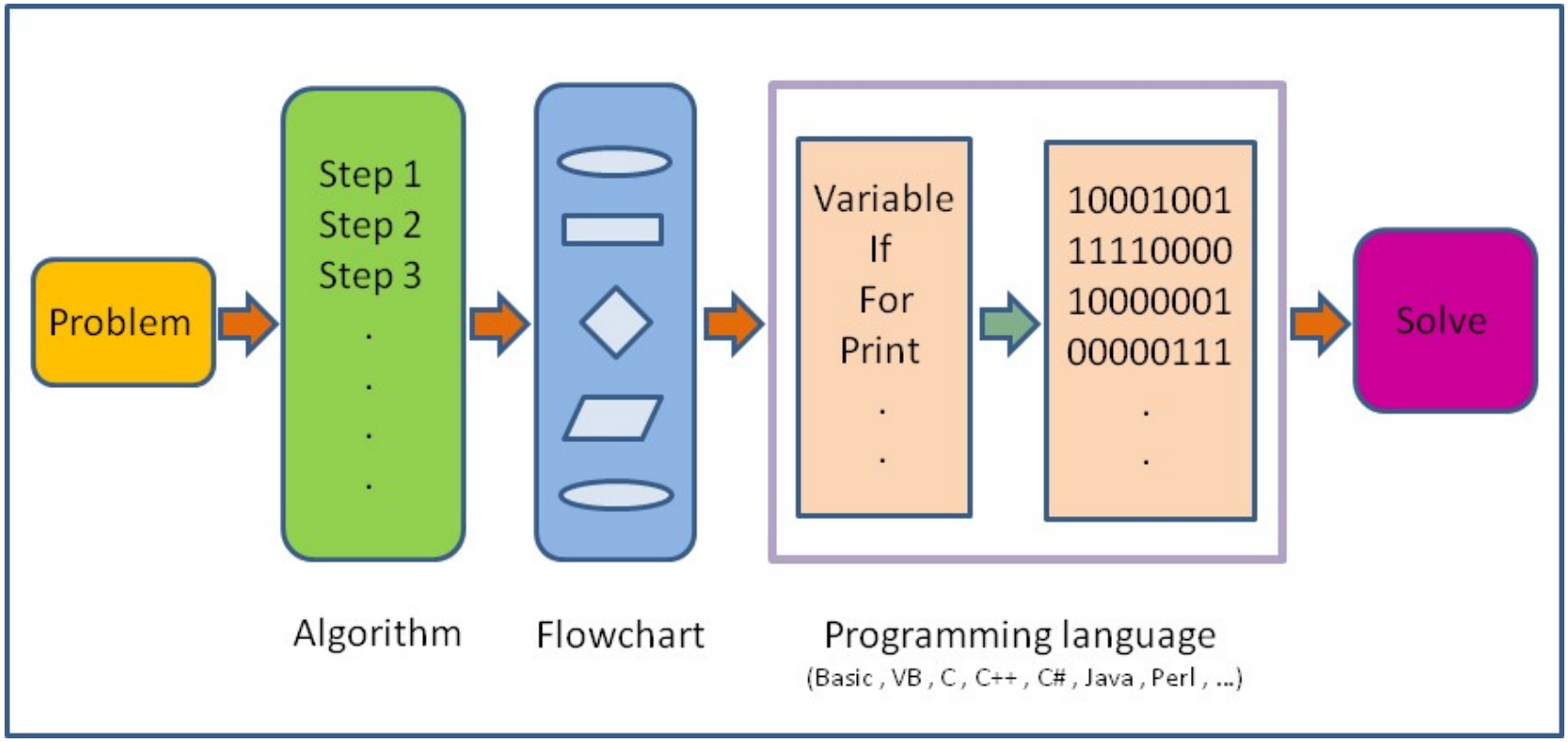


Output

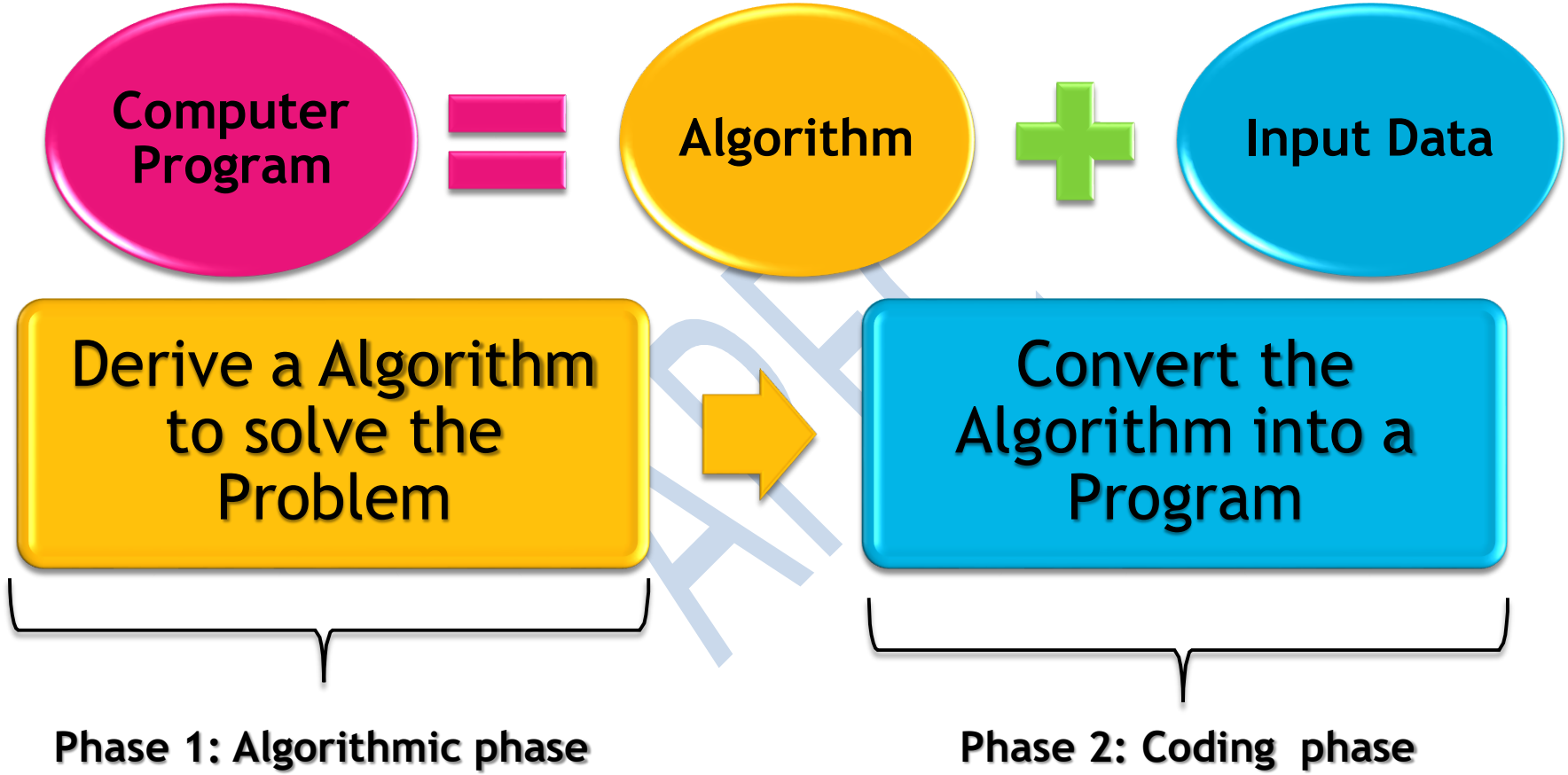
Contd…



Steps

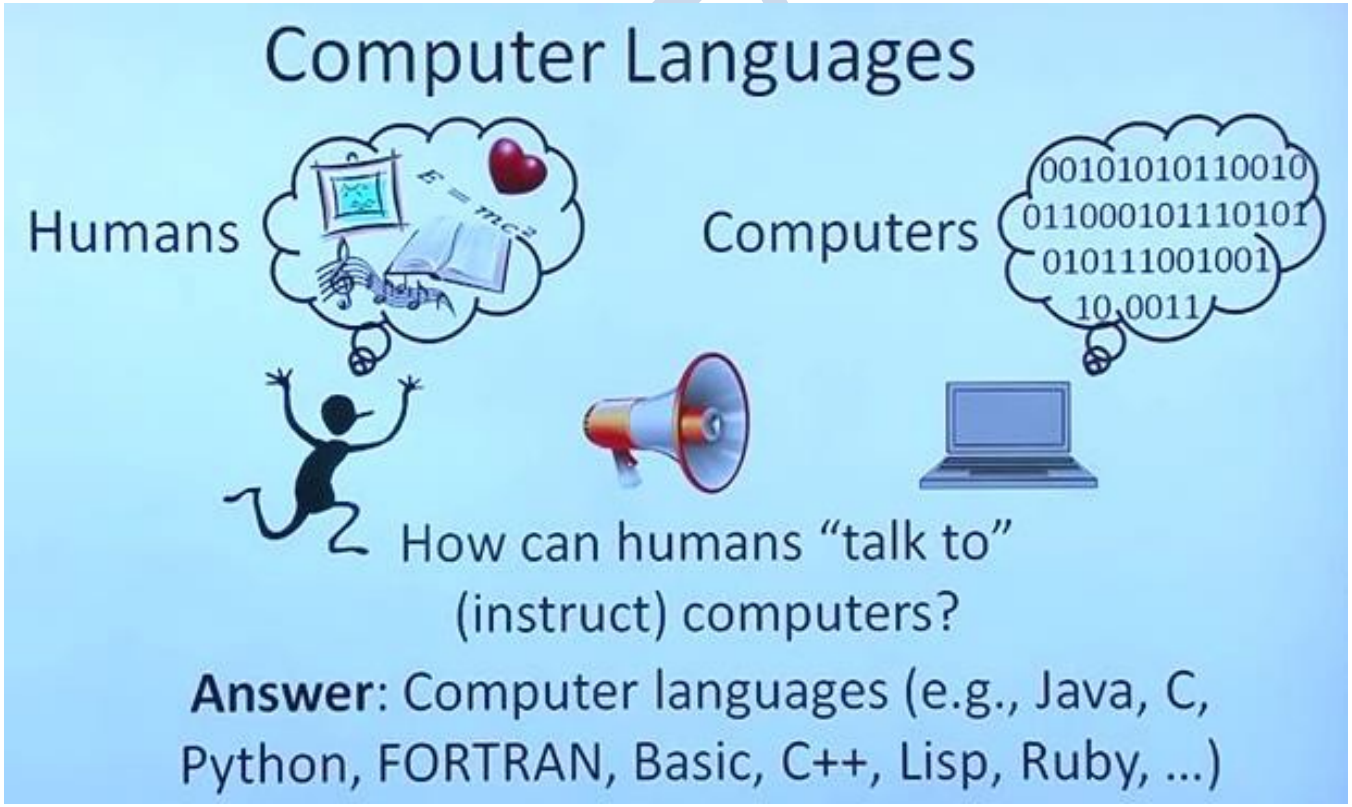

Problem
Step 1
Step 2
Step 3
.
.
.
.
Variable
If
For
Print
.
.
10001001
11110000
10000001
00000111
.
.
Solve
Algorithm
Flowchart
Programming language
(Basic , VB , C , C++ , C# , Java , Perl , ...)

Phases Contd…

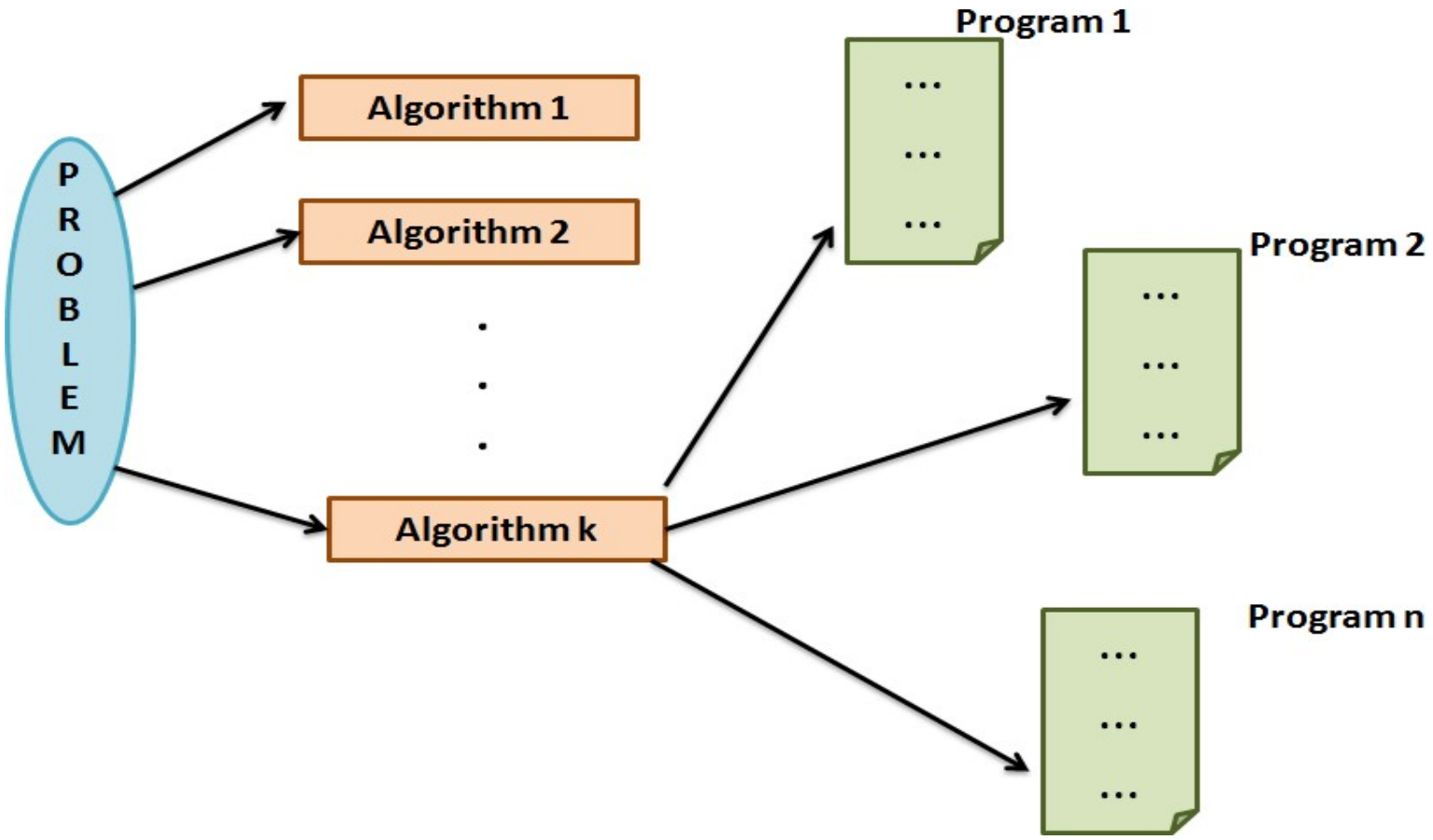


Computer Languages

- Humans talk to computers with the help of Computer languages.



Problems vs Algorithms vs Programs



Algorithm:
Calling a friend on the telephone

- Input: The telephone number of your friend.
- Output: Talk to your friend



- **Steps:**
  1. Pick up the phone and listen for a dial tone
  2. Press each digit of the phone number on the phone
  3. If busy, hang up phone, wait 2 minutes, jump to step 2
  4. If no one answers, leave a message then hang up
  5. If no answering machine, hang up and wait 2 hours, then jump to step 2
  6. Talk to friend
  7. Hang up phone

AREC

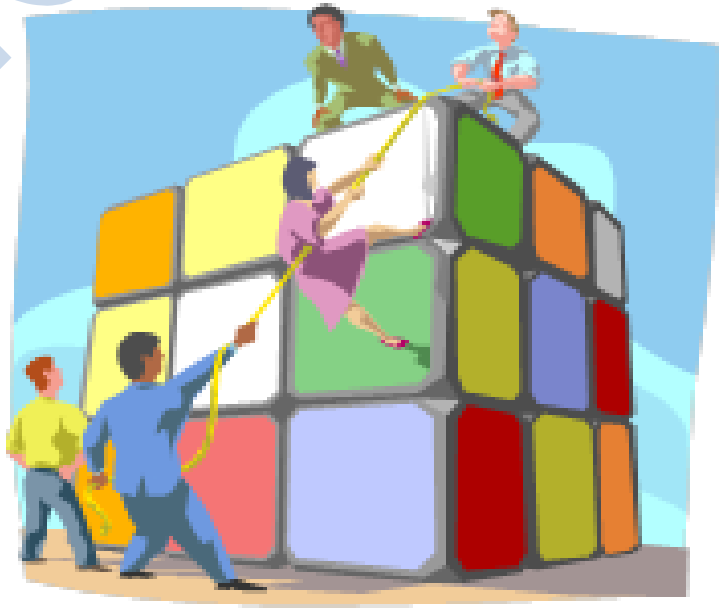
An Algorithm Development Process

- The algorithm development process consists of five major steps.
  - **Step 1:** Obtain a description of the problem.
  - **Step 2:** Analyse the problem.
  - **Step 3**: Develop a high-level algorithm.
  - **Step 4:** Refine the algorithm by adding more detail.
  - **Step 5:** Review the algorithm.

APC

Step 1: Obtain a description of the problem

- The problem should be clearly explained
  - so that it's easy for the developer to find the solution for the problem.
- Not as easy as it seems.



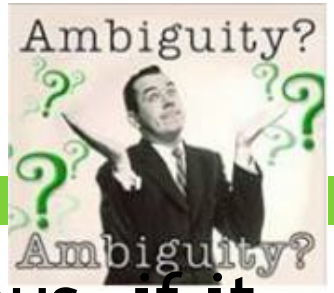
APEC

Step 1: Obtain a description of the problem

- **Common defects are:**
  - Description relies on unstated assumptions.
  - Description is ambiguous.
  - Description is incomplete.
  - Description has internal contradictions.
    - These flaws are seldom due to carelessness by the client. or sometimes by natural languages (English, French, Korean, etc.)

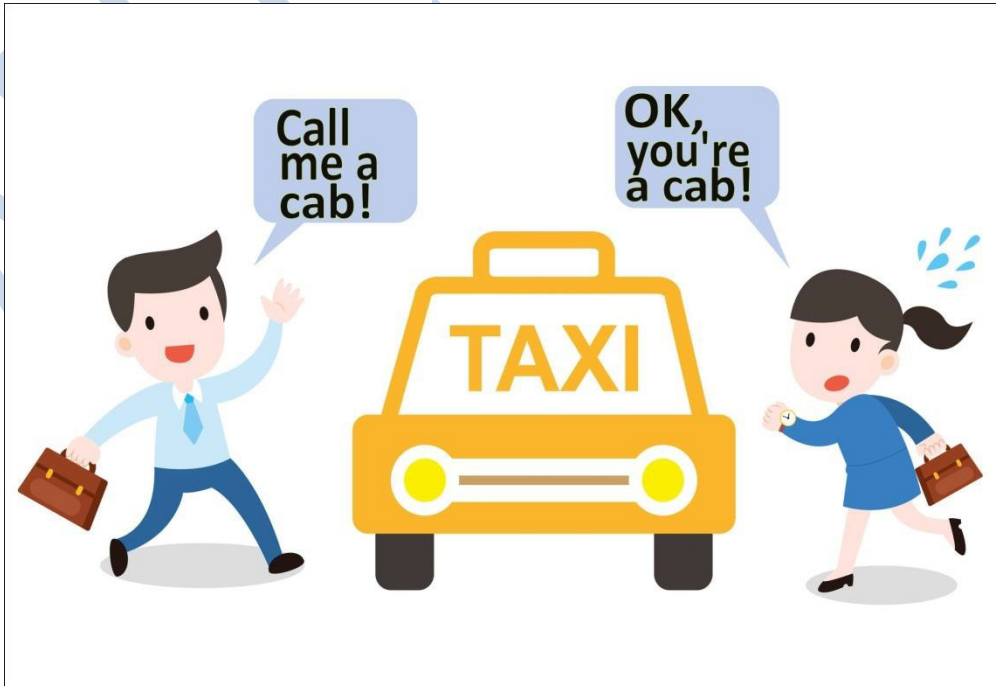
APEC

Description is ambiguous.



- A word, phrase or sentence is ambiguous, if it has more than one meaning (or) has more than one interpretation.

CHIP
- a small piece of wood
- a long thin piece of potato
- a small piece of silicon



Step 2: Analyse the problem.

- Determine both the starting and ending points for solving the problem.



- When determining the starting point, start with following questions:
  - What data are available?
  - Where is that data?
  - What formulas are related to the problem?
  - What rules are needed for the data?
  - What relationships exist among the data values?

APEC

Contd…

- When determining the ending point, ask these following questions.
  - What new facts will arrive?
  - What items will change?
  - What things will no longer exist?

APEC

Step 3: Develop a high-level Algorithm.

- An **algorithm** is a plan for solving a problem.
  - Start with a high-level algorithm ➔ solves major part of the solution.

Example

- **Step 1:** Problem Statement
  - I need a make a tea.



- **Step 2:** Analysis
  - I don't have milk.

- **Step 3:** High-level algorithm
  - Go to a stores that sells milk
  - Purchase milk and come home
  - Prepare Tea

APEC

Contd…

- Though this algorithm seems to be satisfactory, it lacks many details such as the following.
  - Which store I need to visit?
  - Which milk product I need to buy?
  - How I go to the stores: walk, drive, ride my two-wheeler, take the bus.
- These kinds of details are considered in the next step of our process.

APEC

Step 4: Refine the algorithm by adding more detail.

- A high-level algorithm → give major steps to solve a problem.
- Aim:
- Develop algorithm → that lead to a computer programs.
  - So, all details to be considered.

High level → Detailed level

- **Stepwise refinement = Keep on adding details**

APEC

Stepwise refinement

Stepwise refinement is a process for developing a detailed algorithm by gradually adding detail to a high-level algorithm.



Step 5: Review the algorithm.

- Check the algorithm step by step to determine whether it will solve the original problem or not.
  - Does this algorithm solve a very specific problem or does it solve a more general problem?
  - If it solves a very specific problem, should it be generalized?

Example:

- Compute the area of a circle having radius 5.2 meters.
  - If the algorithm have the formula, $\pi * 5.2^2$ then it solves a very specific problem.
  - But, the algorithm computes the area of any circle, $\pi * r^2$then it solves a more general problem.

APEC

Properties of an Algorithm

- **Finiteness:**
  - The algorithm must always terminate after a finite number of steps.
- **Definiteness:**
  - Each instruction must be clear, well-defined and precise. There should not be any ambiguity.
- **Effectiveness:**
  - Each Instruction must be simple and be carried out in a finite amount of time.

APEC

Properties – Contd…

- **Input:**
  - An algorithm has zero or more inputs.
- **Output:**
  - An algorithm has one or more outputs, which have a specified relation to the inputs.
- **Feasibility:**
  - It must be possible to perform each instruction.
- **Generality :**
  - the algorithm must be able to work for a set of inputs rather than a single input.

APEC

Building blocks of algorithms

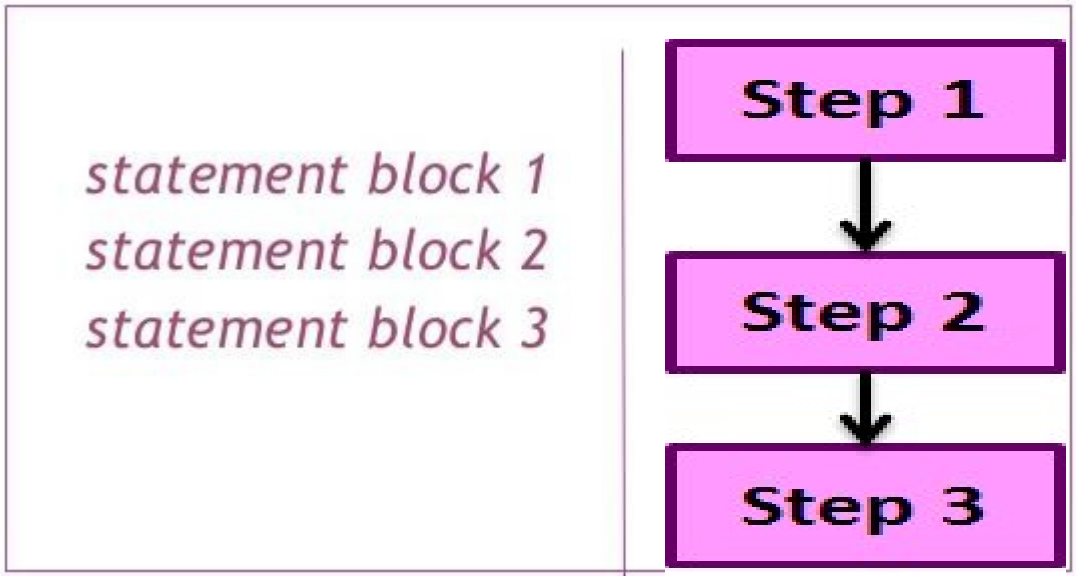
- The three basic building blocks are
  - Sequences
  - Conditionals
  - Loops

- All problems can be solved by using any one of the building blocks or their combinations.

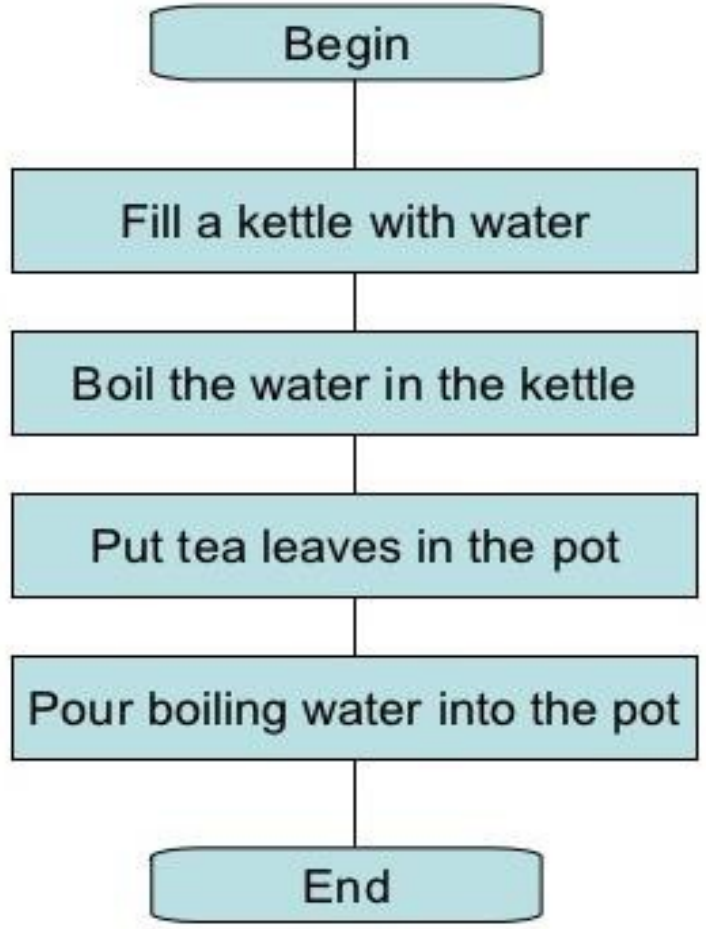
APEC

Sequences

- Sequences are the set of instructions that are executed in the precise order.



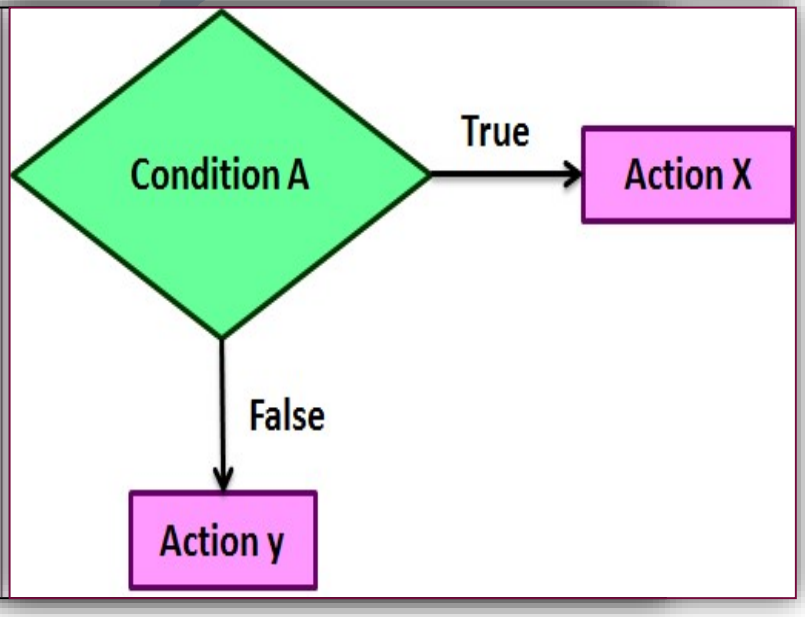
Example – Preparing Tea



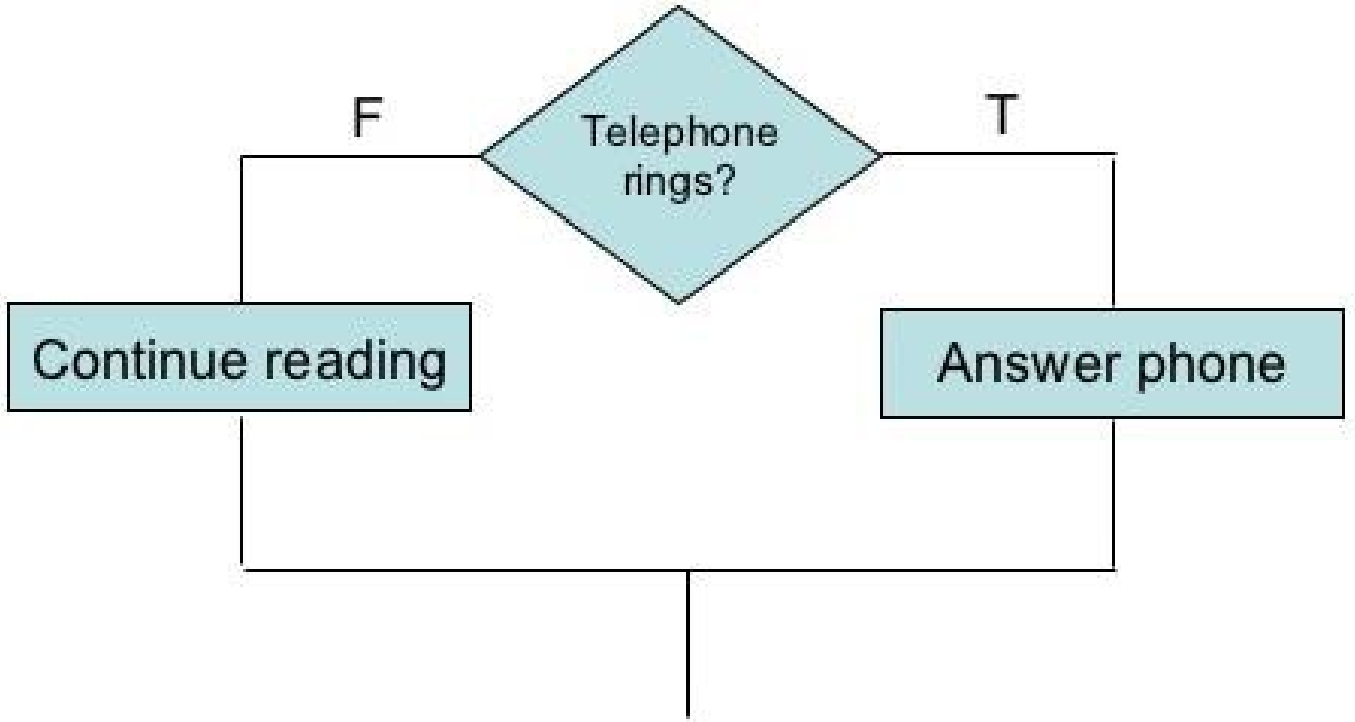
Conditionals

- Selection is a decision or question.

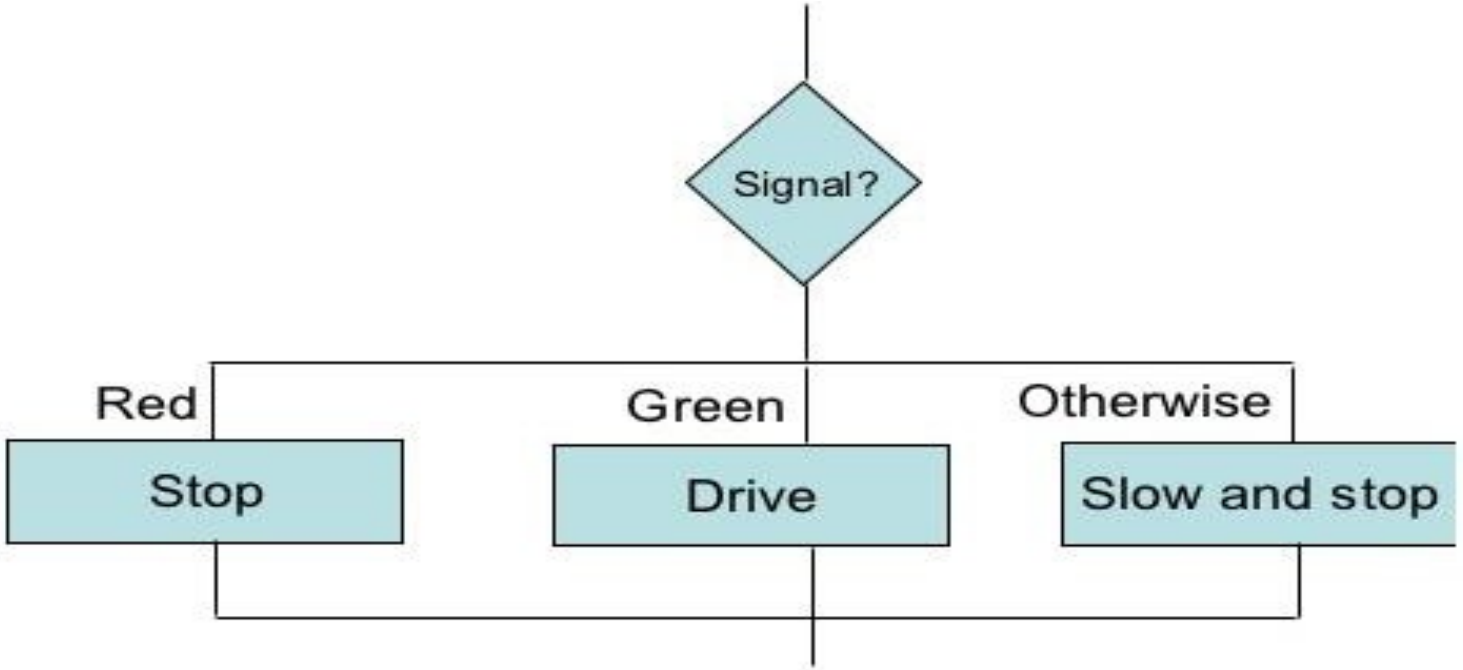
```
If ( condition = true )
     statement block 1
Else
     statement block 2
End if
```



Example – Telephone Ringing

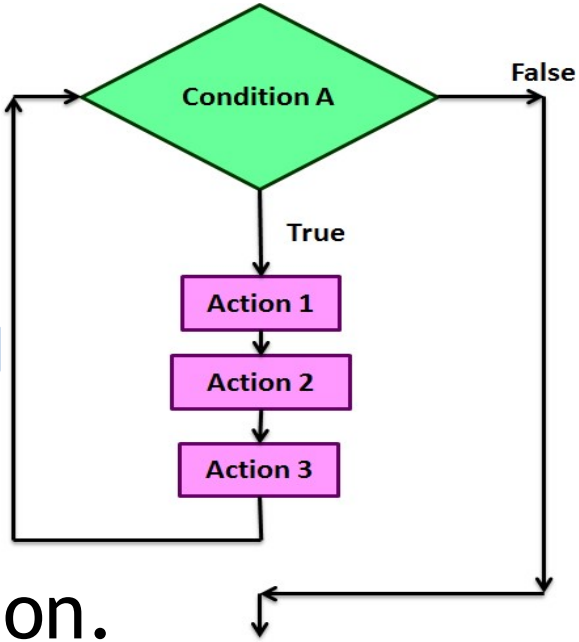


Example 2



Iteration

- Iteration ➔ Used to repeat a set of instructions.



- There are two types of iteration.
  - Count controlled iteration
  - Condition controlled iteration

APEC

Example

Without Iteration

Step 1: Put toothpaste on toothbrush
Step 2: Clean tooth 1 using toothbrush
Step 3: Clean tooth 2 using toothbrush
Step 4: Clean tooth 3 using toothbrush
Step 5: Clean tooth 4 using toothbrush
Step 6: Clean tooth 5 using toothbrush
Step 7: Clean tooth 6 using toothbrush
Step 8: Clean tooth 7 using toothbrush
Step 9: Clean tooth 8 using toothbrush
Step 10: Clean tooth 9 using toothbrush
Step 11: Clean tooth 10 using toothbrush
Step 12: Rinse toothbrush

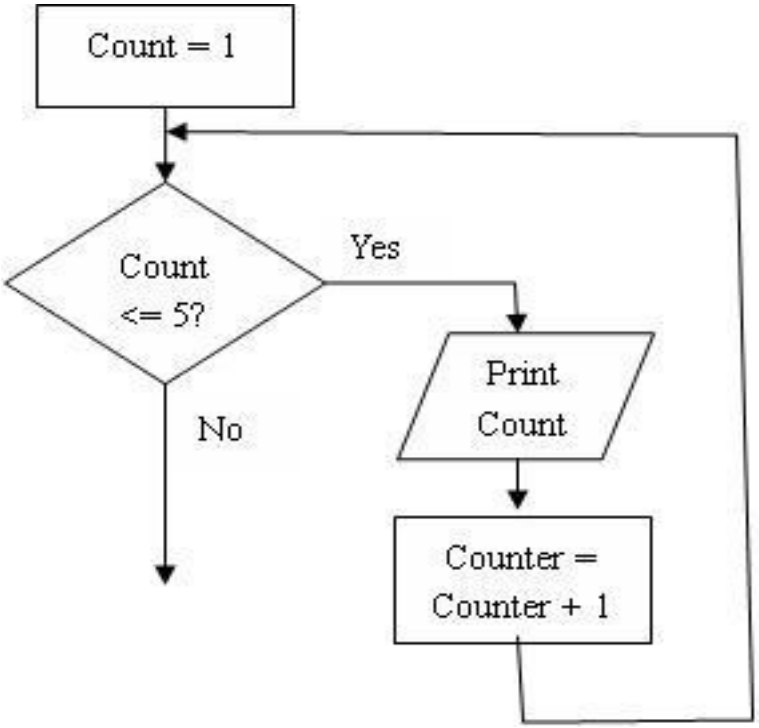
With Iteration

Step 1: Put toothpaste on toothbrush
Step 2: Clean tooth 1 using toothbrush
Step 3: Move to next tooth
Step 4: Repeat steps 2 and 3 until all teeth are clean
Step 5: Rinse toothbrush

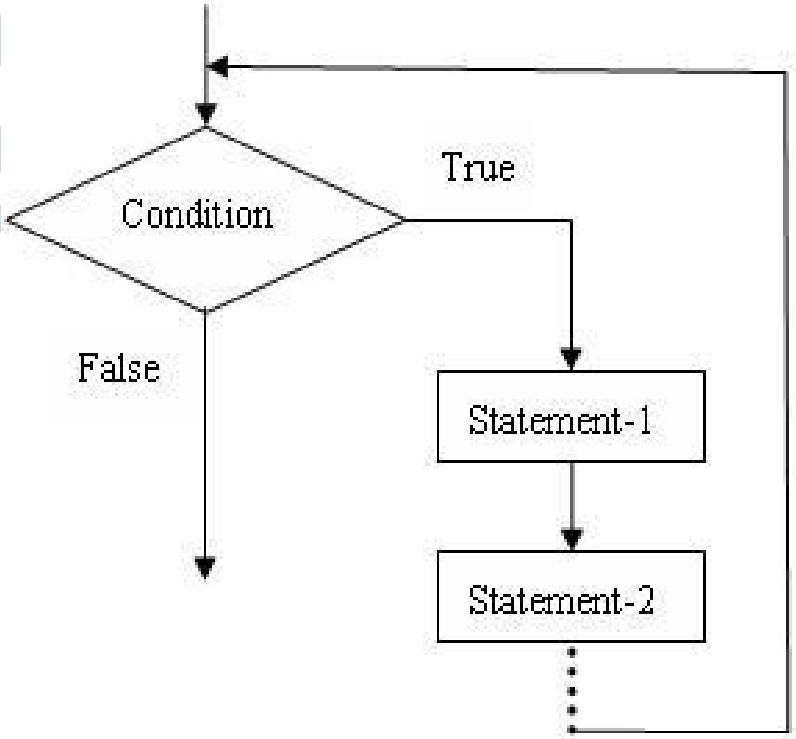
APEC

Iteration - Types

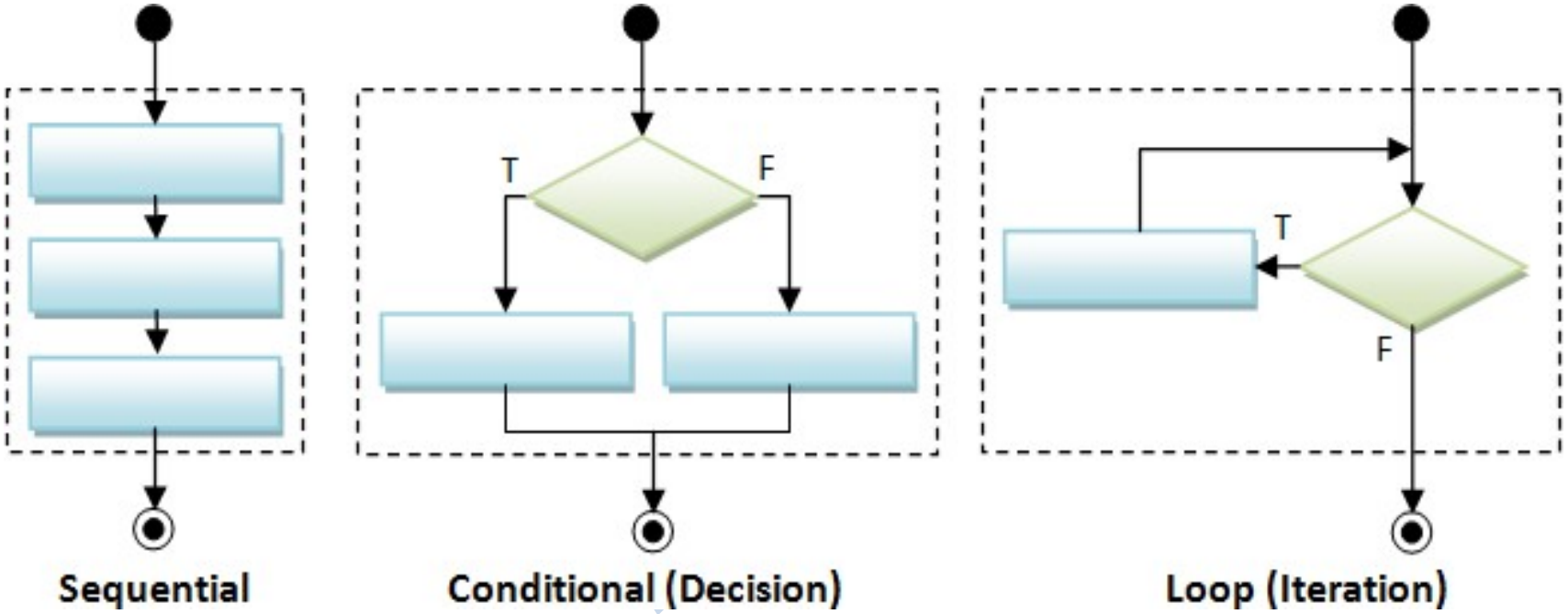
Count controlled iteration



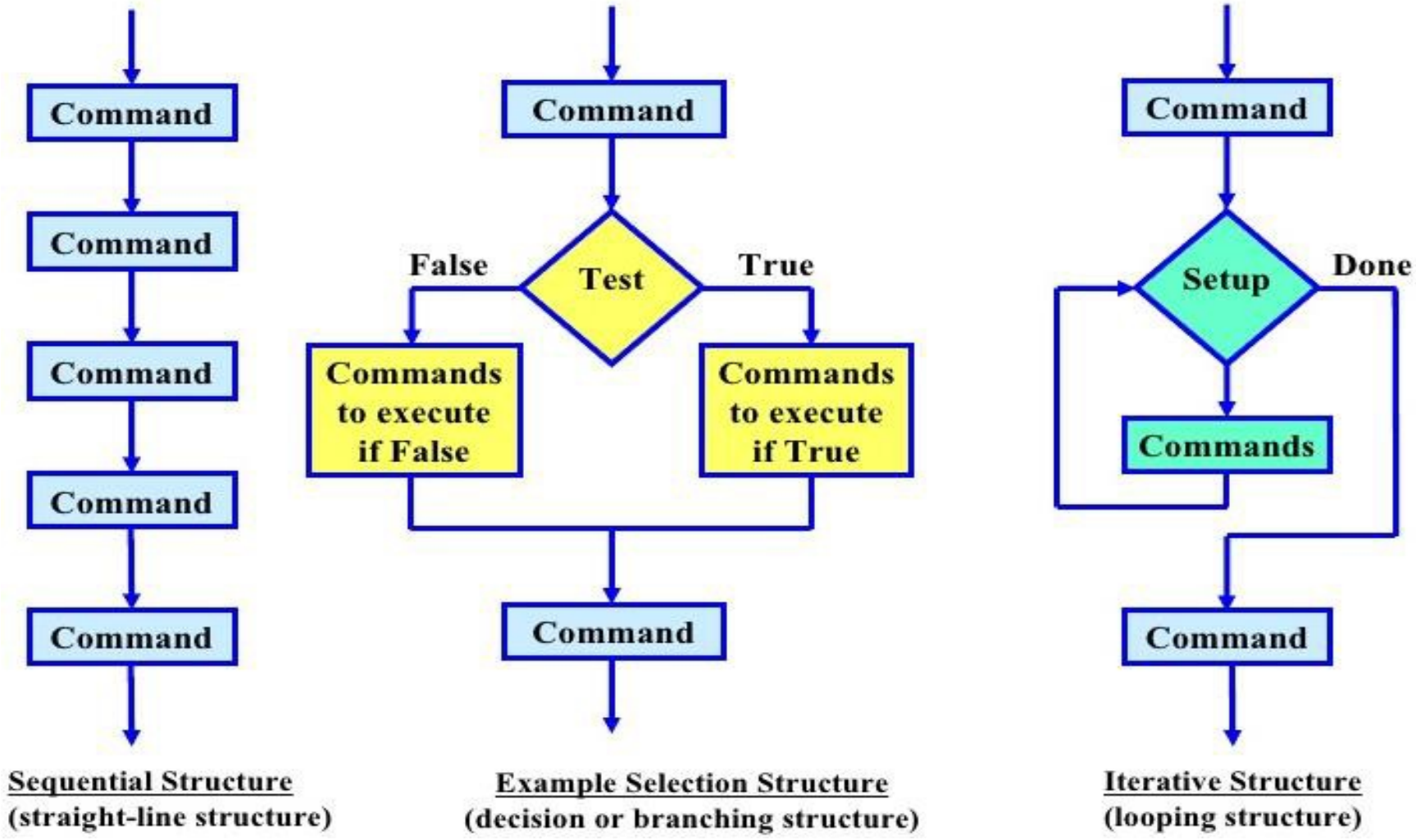
Condition controlled iteration



Contd...



Contd…


Command
Command
Command
Command
Command
Sequential Structure
(straight-line structure)
Command
False
Test
True
Commands to execute if False
Commands to execute if True
Command
Example Selection Structure
(decision or branching structure)
Command
Setup
Done
Commands
Command
Iterative Structure
(looping structure)

FUNCTIONS

Functions

- Modular programming: breaking a program up into smaller, manageable functions or modules
- **Function:** a collection of statements to perform a task

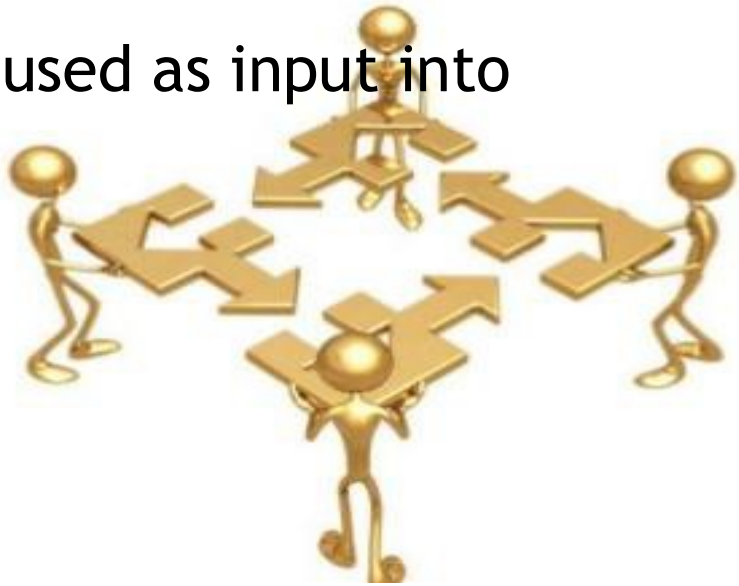
- Motivation for modular programming
  - Improves maintainability of programs
  - Simplifies the process of writing programs

APC

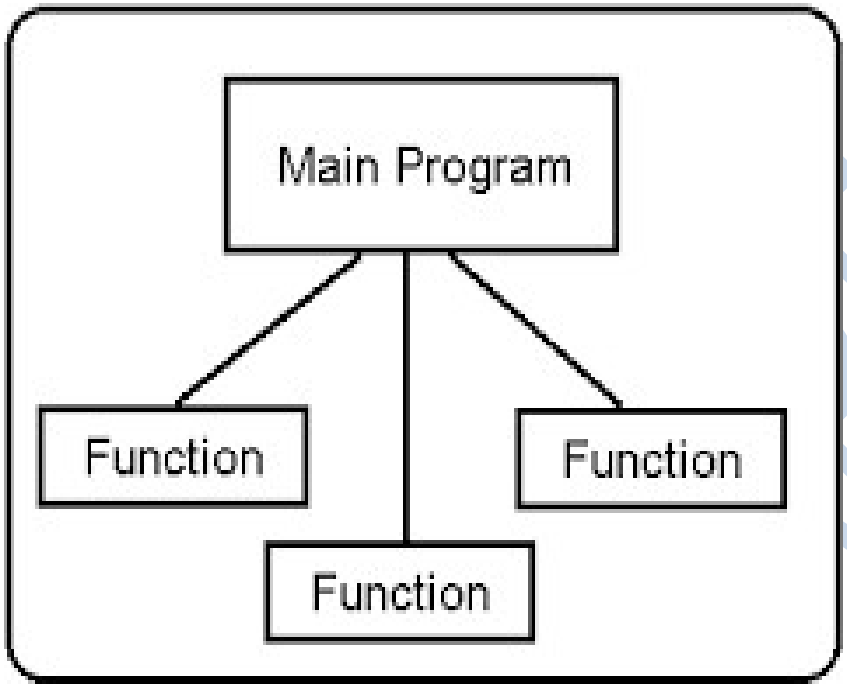
What is Modularity?

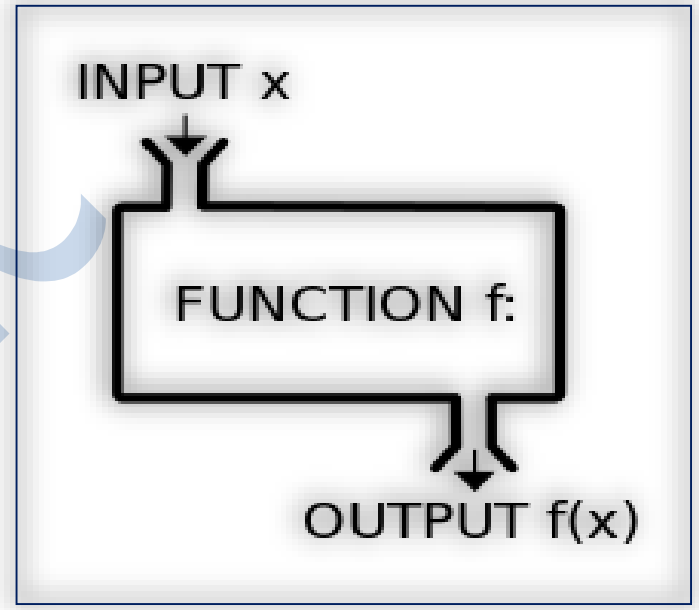
Modularity – Breaking the program into smaller parts

- Each team member has their own module
- Each module has Inputs/Outputs
- Each module does something different
- Outputs from one module are used as input into another module



Function – Contd…





Examples for Algorithms

Write an algorithm to add two numbers entered by user.

- Step 1: Start
- Step 2: Read values num1 and num2.
- Step 3: Add num1 and num2 and assign the result to sum.

  sum = num1+num2
- Step 4: Display sum
- Step 5: Stop

APEC

NOTATIONS

Expressing Algorithms

- Algorithms can be expressed in many different notations
  - **Natural Language**
    - Natural language expressions of algorithms tend to be verbose and ambiguous, and are rarely used for complex or technical algorithms.
  - **Pseudo code**
  - **Flowcharts**
    - Pseudo-code and flowcharts are structured ways to express algorithms that avoid many of the ambiguities common in natural language statements.
  - **Programming languages.**
    - Implementing the algorithms in a form that can be executed by a computer

APEC

Pseudo code

- Pseudocode is an informal language used by programmers to develop algorithms.
- Pseudocode is a "text-based" detail design tool.
  - “Pseudo” ➔ imitation or false
  - “Code” ➔ Computer Instructions

- Algorithm ➔ form the stage 1
- Pseudocode or Flowcharts➔ form the stage 2

APEC

Rules to follow

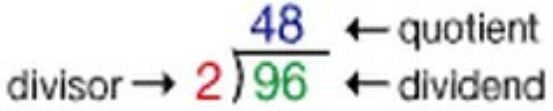
- **Only one statement per line**
  - Readability improves if just one action for the computer is written in one statement
- **Capitalized initial keyword**
  - Keywords like READ, WRITE, etc are in caps.
- **Indent to show hierarchy.**
  - In loops, states and iterations the logically dependent statements must be indented
- **End multi-line structures.**
  - To improve readability the initial start and end of the several lines must be specified properly
- **Keep statement language independent.**
  - The programmer must never use the syntax of any programming language

APEC

Express an algorithm to get two numbers from the user (dividend and divisor), testing to make sure that the divisor number is not zero, and displaying their quotient using pseudocode

1. Declare variables: dividend, divisor, quotient
2. Prompt user to enter dividend and divisor
3. Get dividend and divisor
4. IF divisor is equal to zero, THEN
    4.1. DO
        4.1.1. Display error message, “divisor must be non-zero”
        4.1.2. Prompt user to enter divisor
        4.1.3. Get divisor
    4.2. WHILE divisor is equal to zero
5. ENDIF
6. Display dividend and divisor
7. Calculate quotient as dividend/divisor
8. Display quotient

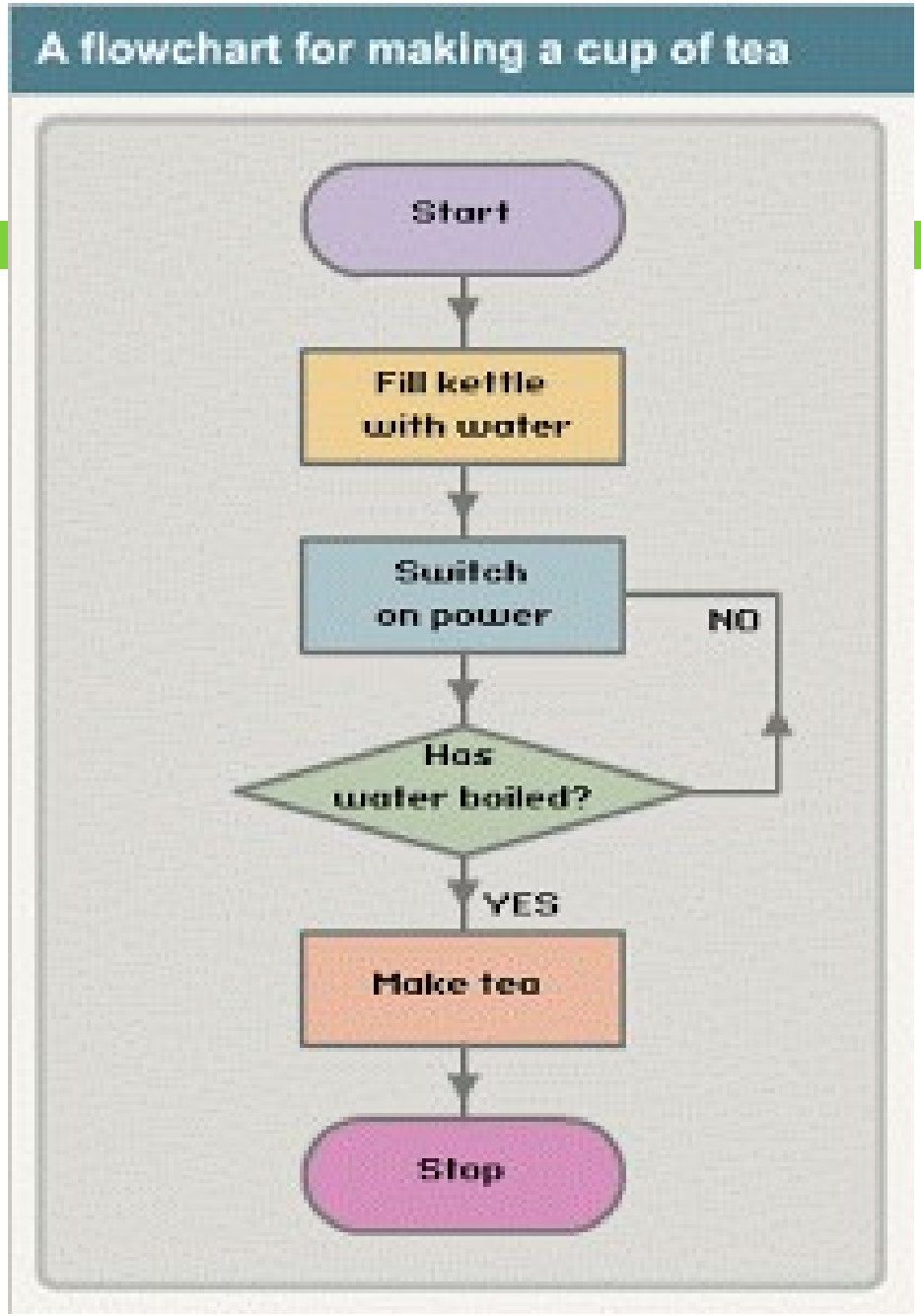




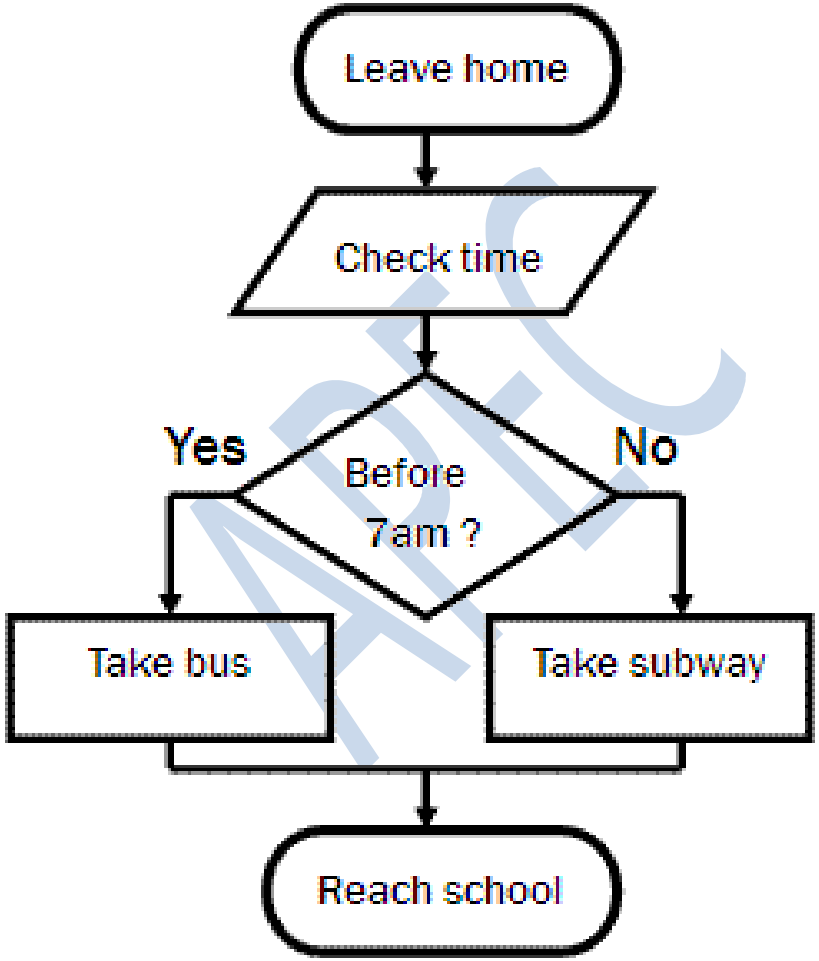
Flowcharts

- A flowchart is a graphical representation of an algorithm.
  - Flowcharts were introduced by Frank Gilberth in 1921.
- A flowchart is a diagram made up of boxes, diamonds and other shapes, connected by arrows
  - Each shape represents a step in the process, and the arrows show the order in which they occur.

APEC

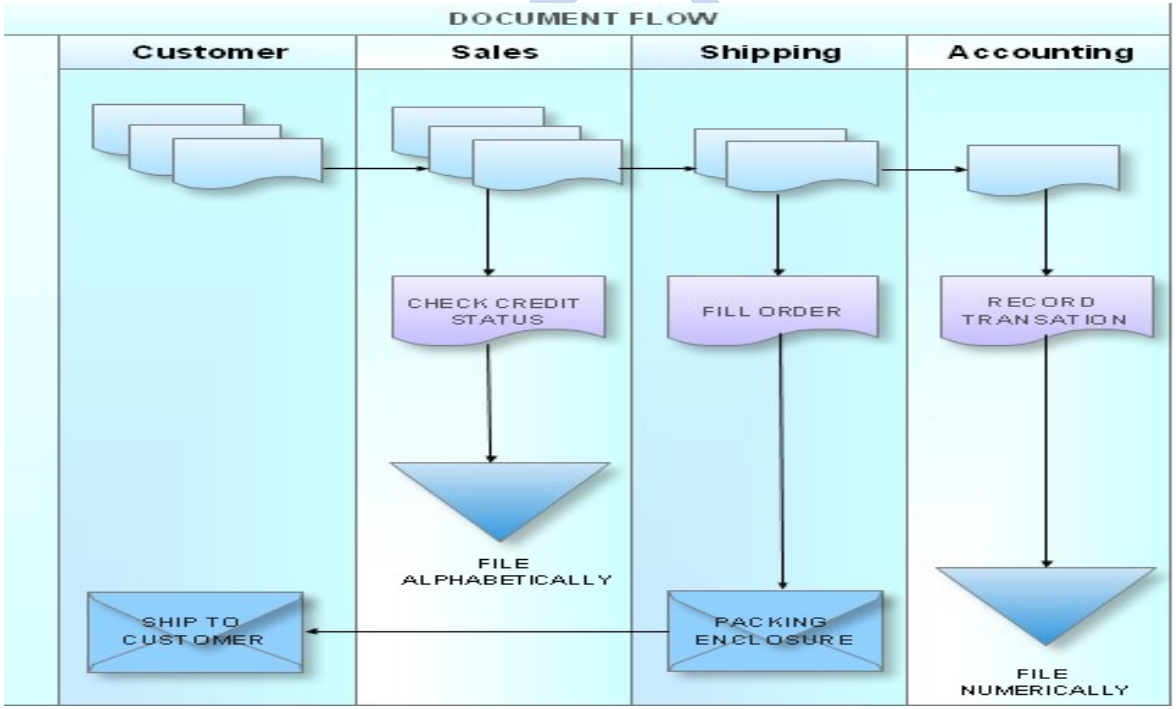

A flowchart for making a cup of tea
Start
Fill kettle with water
Switch on power
NO
Has water boiled?
YES
Make tea
Stop

Example 2


Leave home
Check time
Yes
Before
7am ?
No
Take bus
Take subway
Reach school

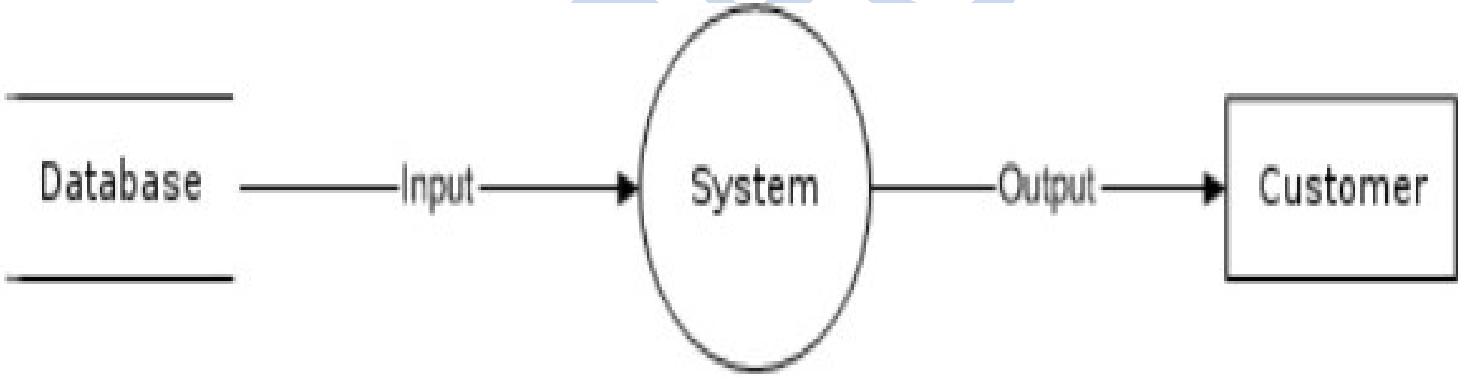
Types of Flowcharts

- **Document flowcharts**
  - Explain the document-flow through the components of the system
  - This chart is read from left to right and explain the flow of documents through various business units.



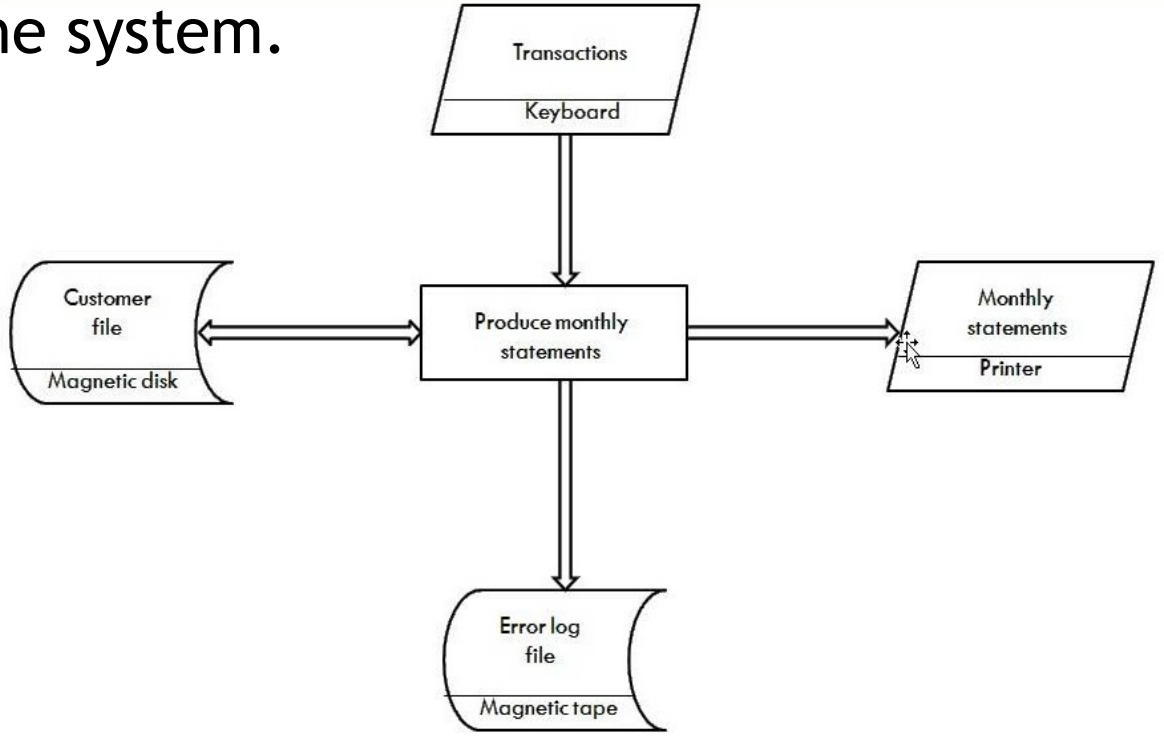
Types of Flowcharts – Contd…

- **Data flowcharts**
  - Used primarily to show how data is transmitted through the system rather than the control flow.



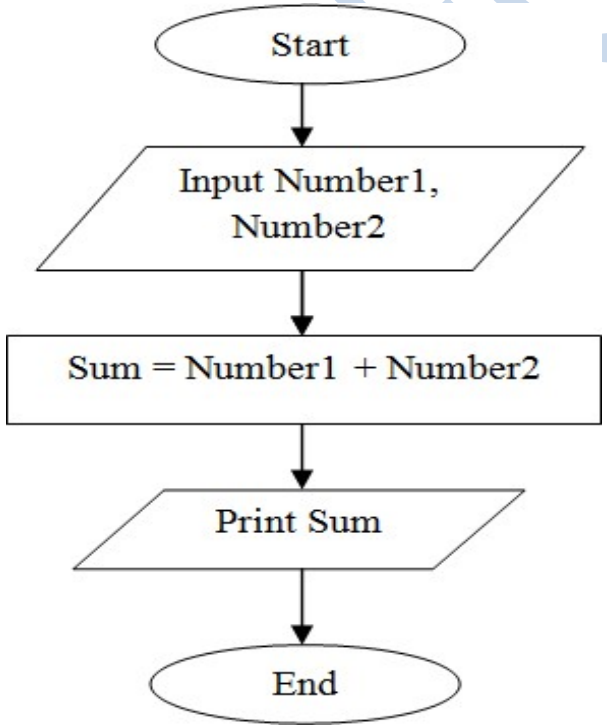
Types of Flowcharts – Contd…

- **System flowcharts**
  - Specify how controls are located at the physical or resource level.
  - It shows the flow of data through the major components of the system.



Types of Flowcharts

- **Program flowchart**
  - shows the controls placed internally to the program within a system.
  - Program flowchart is a gift to programmers as it makes programming task very easy and systematic



Flowcharting Symbols

- Process / Operation Symbols
- Branching and Control of Flow Symbols
- Input and Output Symbols
- File and Information Storage Symbols
- Data Processing Symbols

APEC

Process / Operation Symbols

Process / Operation Symbols in Flowchart		
Symbol	**Name**	**Description**
	Process	Show a Process or action step. This is the most common symbol used in flowchart.
	Predefined Process (Subroutine)	A Predefined Process symbol is a marker for another process or series of process. This shape commonly depicts sub-processes or subroutines.
	Alternate Process	It is used when the process flow step is an alternate to the normal process step.
	Delay	The Delay flowchart symbol depicts any waiting period that is part of a process.
	Preparation	Any process step that is a Preparation process flow step, such as a set-up operation.
	Manual Operation	Manual Operations flowchart shapes show which process steps are not automated.

APC

Branching and Control of Flow Symbols in Flowchart		
Symbol	Name	Description
	Flow Line (Arrow, Connector)	Flow line connectors show the direction that the process flows.
	Terminator (Terminal Point, Oval)	Terminators show the start and stop points in a process.
	Decision	Indicates a question or branch in the process flow.
	Connector (Inspection)	It is used to show a jump from one point in the process flow to another.
	Off-Page Connector	Off-Page Connector shows continuation of a process flowchart onto another page.
	Merge (Storage)	This symbol shows the merging of multiple processes or information into one.
	Extract (Measurement)	This symbol shows when a process splits into parallel paths.
	Or	The logical Or symbol shows when a process diverges - usually for more than 2 branches.
	Summing Junction	The logical Summing Junction flowchart shape is shows when multiple branches converge into a single process.

File Symbols

File and Information Storage Symbols in Flowchart		
Symbol	**Name**	**Description**
	Stored Data	A general Data Storage flowchart shape used for any process step that stores data.
	Magnetic Disk (Database	The most universally recognizable symbol for a data storage location, this flowchart shape depicts a database.
	Direct Access Storage	Direct Access Storage is a like Hard Drive.
	Internal Storage	It is used in programming flowcharts to store information in memory.
	Sequential Access Storage (Magnetic Tape)	Although it looks like a 'Q', the symbol is supposed to look like a reel of tape.

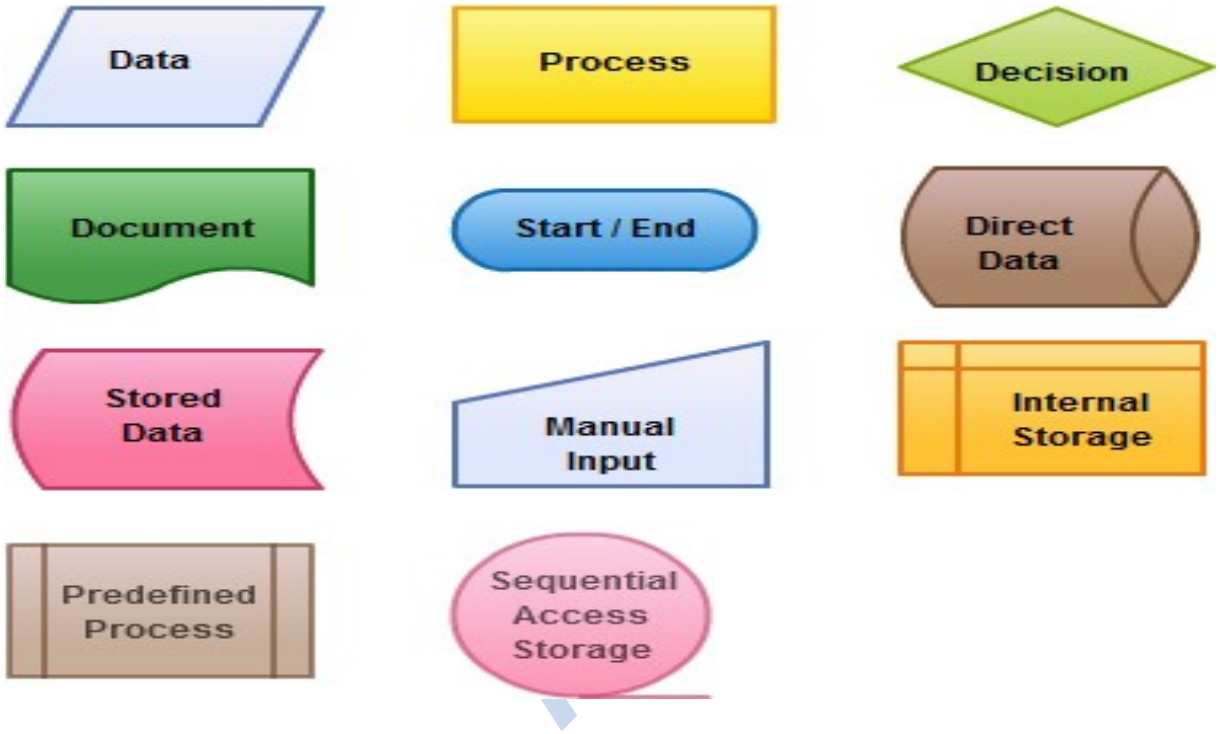
APEC

Data Processing Symbols

Data Processing Symbols in Flowchart		
Symbol	**Name**	**Description**
	Collate	The Collate flowchart shape indicates a process step that requires organizing data or information according to a standard format.
	Sort	Indicates the sorting of data, information, materials into some pre-defined order.

APEC

Frequently used symbols



Guidelines for drawing

or

Flowcharts

APEC

- All necessary requirements should be listed out

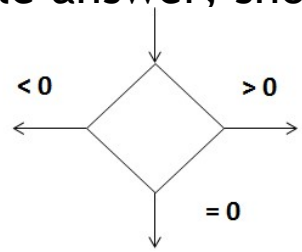
in logical order for drawing a proper flowchart.

- The flowchart should be clear, neat and easy to follow.
- The usual direction of the flow of a process is from left to right or top to bottom.
- Only one flow line should come out from a process symbol.

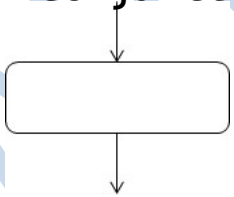


Contd…

- Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, should leave the decision symbol.



- Only one flow line is used in conjunction with terminal symbol.



- If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines.
- Ensure that the flowchart has a logical *start* and *finish*.
- It is useful to test the validity of the flowchart by passing through it with a simple test data.

Advantages of using Flowcharts

- **Communication:**
  - Flowcharts are better way of communicating the logic of a system.
- **Effective analysis**
  - With the help of flowchart, problem can be analysed in more effective way.
- **Proper documentation**
  - Flowcharts are used for good program documentation, which is needed for various purposes.
- **Efficient Coding**
  - Flowcharts act as a guide or blueprint during the systems analysis and program development phase.
- **Proper Debugging**
  - Flowchart helps in debugging process.
- **Efficient Program Maintenance**
  - The maintenance of running program becomes easy with the help of flowchart.

APEC

Limitations of using Flowcharts

- **Complex logic**
  - Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
- **Alterations and Modifications**
  - If alterations are required the flowchart may require re-drawing completely.
- **Reproduction**
  - As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.

APEC

Programming Language

- A programming language is a notation for writing programs, which are specifications of a computation or algorithm

APEC

Qualities of a programming language

- Languages are not designed to provide a means for having a two-way dialog witha computer.
- It is a set of instructions specified by the human on what thecomputer should do.
- Provides a way for humans to communicate to computers
- It always follows grammar; hence there is no ambiguity
- A system will be able to infer the computer language

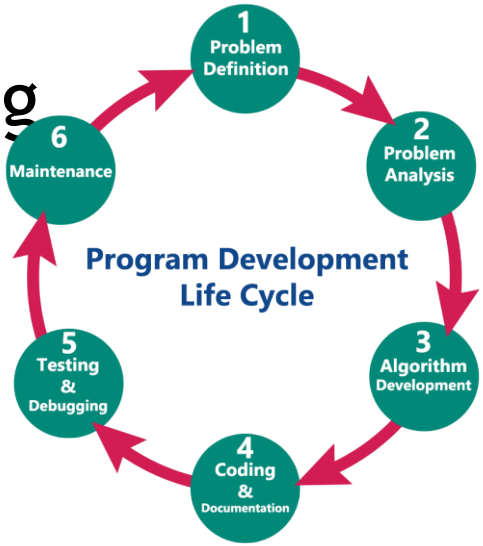
APEC

Steps in Problem Solving

APC

Steps in the problem solving process

- Problem Definition
- Problem Analysis
- Algorithm Development
- Program Coding
- Program Testing and Debugging
- Documentation



APEC

Problem Definition

- Define or specify the problem by answering to the following questions:
  - What the computer program do?
  - What tasks will it perform?
  - What kind of data will it use, and where will it get the data?
  - What will be the output of the program?
  - How will the program interact with the computer user?

APEC

Problem Analysis

- Analysing the problem involves
  - Identifying the problem inputs
  - Check any additional requirements or constraints
  - Determine the required format of the results to be displayed

APEC

Algorithm development

- An algorithm is a finite set of steps defining the solution of a particular problem.
  - The algorithm development can be expressed by
  - **Pseudo code**
    - Pseudo code describes the outline of a program, written in a form that can easily be converted into real programming statements.
  - **Flowchart**
    - Flowchart is the diagrammatic representation of the algorithms

AREC

Program Coding

- Coding or programming is the process of translating the algorithm into the syntax of a given programming language.
  - Convert each step in the algorithm into one or more statements in a programming language.

APEC

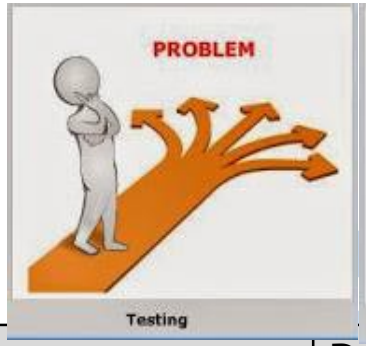
Program Testing and Debugging

- **Program testing**
  - Running the program and executing all its instructions or functions step by step.
- **Debugging**
  - Process of finding and correcting program code mistakes.

- Errors in the program may be
  - Syntax errors
  - Run-time errors
  - Logic errors (or so called bugs)

APEC

Testing Vs Debugging





Testing	Debugging
Finding and locating the defect	Fixing that defect
Done by the Testing team	Done by the Development team
Intention behind is to find as many defect as possible	Intention is to remove those defects

Documentation

- **Internal documentation**
  - Comment statements written in the program itself
- **External documentation**
  - External documentation is made up of the manuals or help menus written about the solution

APEC

Recursion and Iteration

APEC

Iteration

- In iteration, the control statements such as for loop, do-while loop or while loop is used.

- Refer Slide no: 53

APEC

Steps to develop an Iterative Program

- **Define Problem** :
  - The problem that needs an iteration has to be identified and defined
- **Initial Conditions:**
  - The condition that has to satisfied to start the iteration
- **Define Loop Invariants :**
  - The variable that controls the number of iteration has to be defined
- **Define Step :**
  - The steps that are to be repeated must be defined

APEC

Steps to develop an Iterative Program

- **Define Measure of Progress:**
  - The loop invariants that have been defined would be changed when the algorithm progresses. How it progresses has to be defined.
- **Define Exit Condition:**
  - When the iteration should be stopped has to be indentified
- **Make Progress:**
  - Move forward after executing an instruction
- **Maintain Loop Invariants:**
  - In order to repeat the steps the loop invariant must be maintained in a range.
- **Ending:**
  - When the iteration has to stop

APEC

Iteration

```
begin routine
    _pre-cond_
    code pre-loop
        loop
            _loop-invariant _
            exit when _exit-cond_
            Codeloop%
        end loop
    Code post-loop
    _post-cond_
end routine
```

APEC

Recursion

- The process in which a function calls itself repeatedly is called recursion and the corresponding function is called as recursive function.

APEC

Structure of the Recursive program

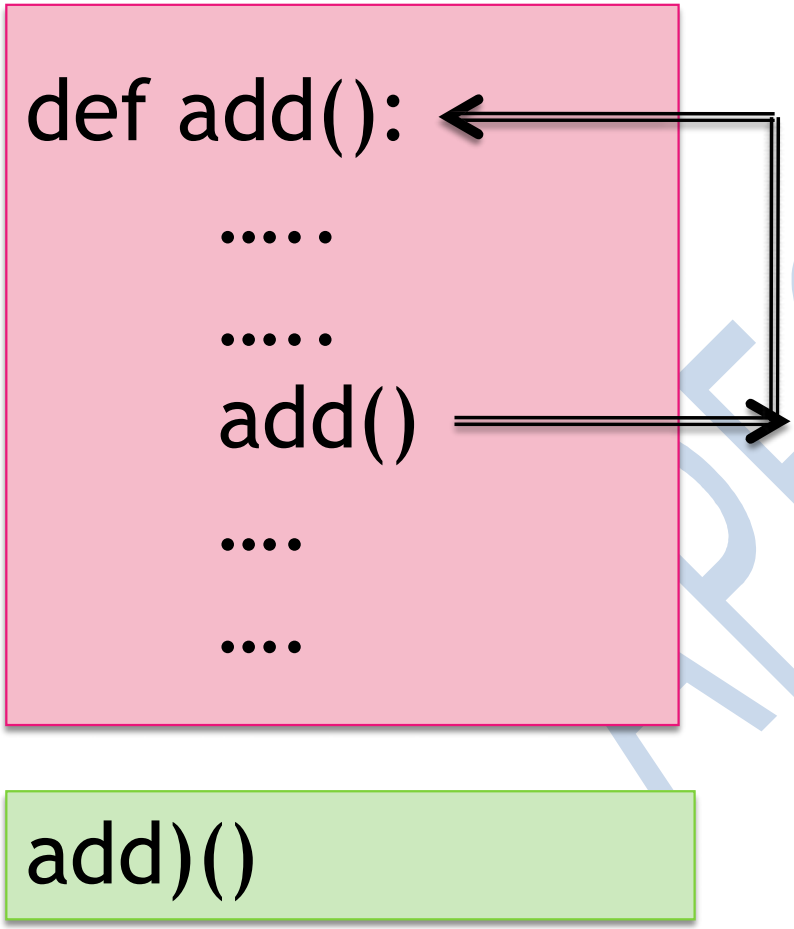
```
def add():
        .....
        .....
        add()
        ....
        ....
```

Function Definition

```
add)()
```

Function Call

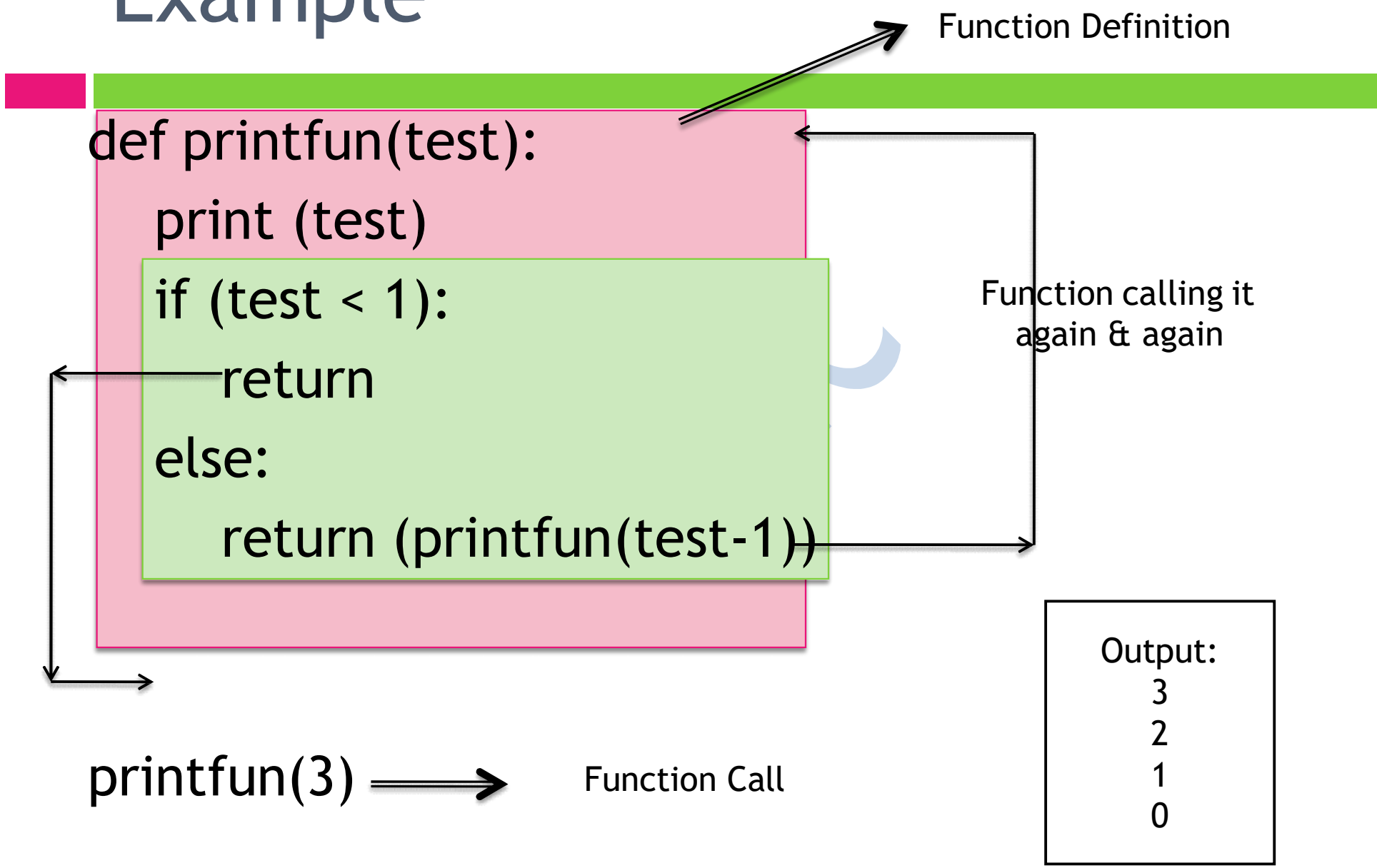
Structure of the Recursive program



Function Calling itself again and again = Recursion

Best Example : Factorial of a given number, Towers of Hanoi

Example


Function Definition
def printfun(test):
print (test)
if (test < 1):
return
else:
return (printfun(test-1))
Function calling it
again & again
printfun(3)
Function Call
Output:
3
2
1
0