UNIT-I

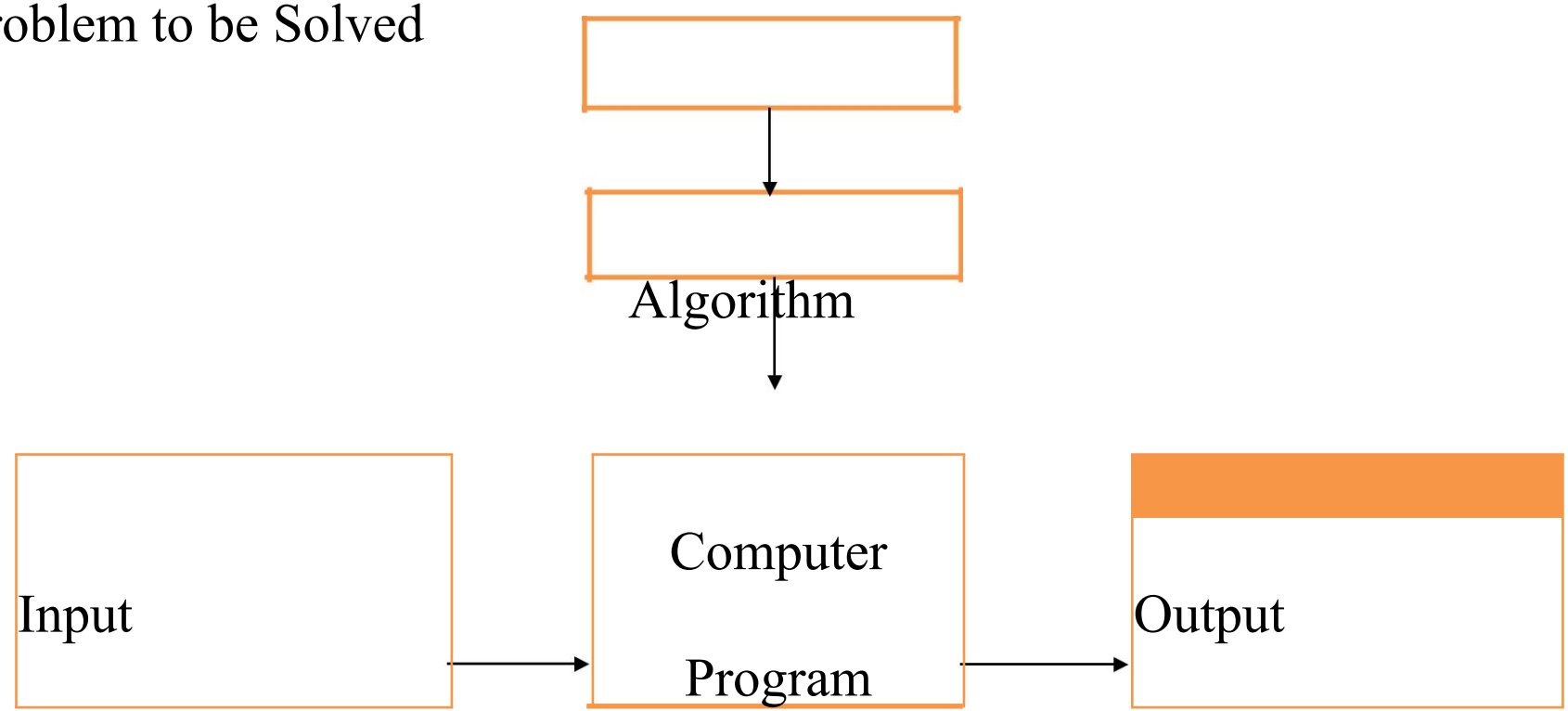
CONTENTS

Introduction to Algorithm:

NOTION OF AN ALGORITHM

An ***algorithm*** is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time

Problem to be Solved



Characteristics of an algorithm:

Input: Zero / more quantities are externally supplied**.**

Output: At least one quantity is produced.

Definiteness: Each instruction is clear and unambiguous.

Finiteness: If the instructions of an algorithm is traced then for all cases the algorithm must terminates after a finite number of steps.

Efficiency: Every instruction must be very basic and runs in short time.

Steps for writing an algorithm:

An algorithm is a procedure. It has two parts; the first part is **head** and the second part is **body**.

The Head section consists of keyword **Algorithm** and Name of the algorithm with parameter list. E.g. Algorithm name1(p1, p2,…,p3). The head section also has the following:

//Problem Description

//Input:

//Output:

In the body of an algorithm various programming constructs like **if, for, while** and some statements like assignments are used.

The compound statements may be enclosed with { and } brackets. **if**, **for**, **while** can be closed by **endif**, **endfor**, **endwhile** respectively. Proper indention is must for block.

The **identifier** should begin by a letter and not by digit. It contains alpha numeric letters after first letter. No need to mention data types.

Input and Output can be done using **read** and **write**.

FUNDAMENTALS OF ALGORIHTMIC PROBLEM SOLVING

A sequence of steps involved in designing and analyzing an algorithm is shown in the figure

(i) Understanding the Problem

This is the first step in designing of algorithm.

Identify the problem types and use existing algorithm to find solution.

Input (***instance***) to the problem and range of the input get fixed.

(ii) Decision making

The Decision making is done on the following:

(iii) Methods of Specifying an Algorithm

There are three ways to specify an algorithm. They are**:**

a. Natural language:

b. Pseudocode:

c. Flowchart:

Algorithmic Specifications

Natural Language	Pseudocode	Flowchart

a. Natural Language

It is very simple and easy to specify an algorithm using natural language. But many times specification of algorithm by using natural language is not clear and thereby we get brief specification.

Example: An algorithm to perform addition of two numbers.

Step 1: Read the first number, say a.

Step 2: Read the first number, say b.

Step 3: Add the above two numbers and store the result in c.

Step 4: Display the result from c.

Such a specification creates difficulty while actually implementing it. Hence many programmers

prefer to have specification of algorithm by means of Pseudocode.

b. Pseudocode

Pseudocode is a mixture of a natural language and programming language constructs. Pseudocode is usually more precise than natural language.

For Assignment operation left arrow "←", for comments two slashes "**//**",**if** condition**, for,while** loops are used.

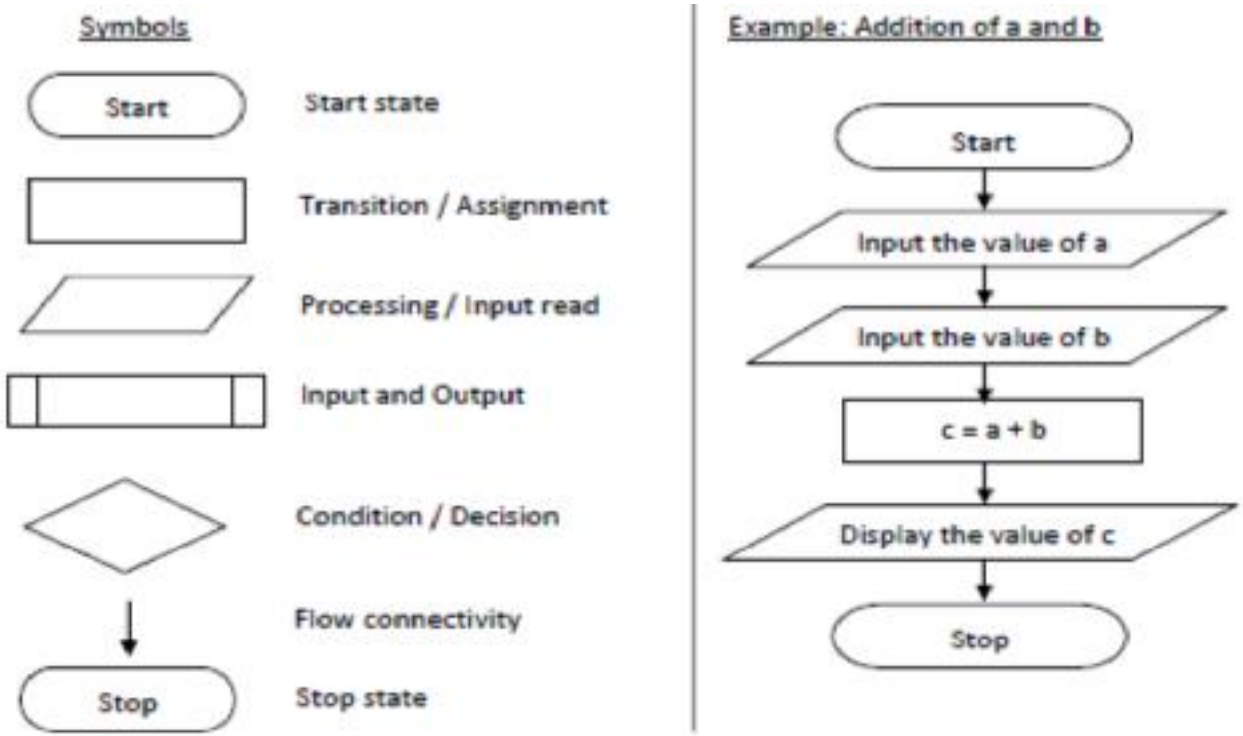
ALGORITHM *Sum(a,b)*

//Problem Description: This algorithm performs addition of two numbers

```
//Input: Two integers a and b
//Output: Addition of two integers
c←a+b
return c
```

c. Flowchart

Flowchart is a graphical representation of an algorithm. It is a a method of expressing an algorithm by a collection of connected geometric shapes
containing descriptions of the algorithm's step



(iv) Proving an Algorithm's Correctness

Once an algorithm has been specified then its ***correctness*** must be proved.

An algorithm must yields a required **result** for every legitimate input in a finite amount of time.

(v) Analyzing an Algorithm

For an algorithm the most important is ***efficiency***. In fact, there are two kinds of algorithm efficiency. They are:

Time efficiency, indicating how fast the algorithm runs, and

Space efficiency, indicating how much extra memory it uses.

The efficiency of an algorithm is determined by measuring both time efficiency and space efficiency.

So factors to analyze an algorithm are:

Time efficiency of an algorithm

Space efficiency of an algorithm

Simplicity of an algorithm

Generality of an algorithm

(vi) Coding an Algorithm

The coding / implementation of an algorithm is done by a suitable programming language like C, C++, JAVA.

It is very essential to write an **optimized code (efficient code)** to reduce the burden of compiler.