INTRODUCTION

The flowchart is a means of visually presenting the flow of data through an information processing systems, the operations performed within the system and the sequence in which they are performed. In this lesson, we shall concern ourselves with the program flowchart, which describes what operations (and in what sequence) are required to solve a given problem. The program flowchart can be likened to the blueprint of a building. As we know a designer draws a blueprint before starting construction on a building. Similarly, a programmer prefers to draw a flowchart prior to writing a computer program. As in the case of the drawing of a blueprint, the flowchart is drawn according to defined rules and using standard flowchart symbols prescribed by the American National Standard Institute, Inc.

OBJECTIVES

At the end of this lesson, you will be able to understand:

- the meaning of flowchart
- the basic parts of the flowchart such as flowchart symbols and the flow lines connecting these symbols.
- the advantages and limitations of flowchart

MEANING OF A FLOWCHART

A flowchart is a diagrammatic representation that illustrates the sequence of operations to be performed to get the solution of a problem. Flowcharts are generally drawn in the early stages of formulating computer solutions. Flowcharts facilitate communication between programmers and business people. These flowcharts play a vital role in the programming of a problem and are quite helpful in understanding the logic of complicated and lengthy problems. Once the flowchart is drawn, it becomes easy to

write the program in any high level language. Often we see how flowcharts are helpful in explaining the program to others. Hence, it is correct to say that a flowchart is a must for the better documentation of a complex program.

GUIDELINES FOR DRAWING A FLOWCHART

Flowcharts are usually drawn using some standard symbols; however, some special symbols can also be developed when required. Some standard symbols, which are frequently required for flowcharting many computer programs are shown in Fig. 25.1
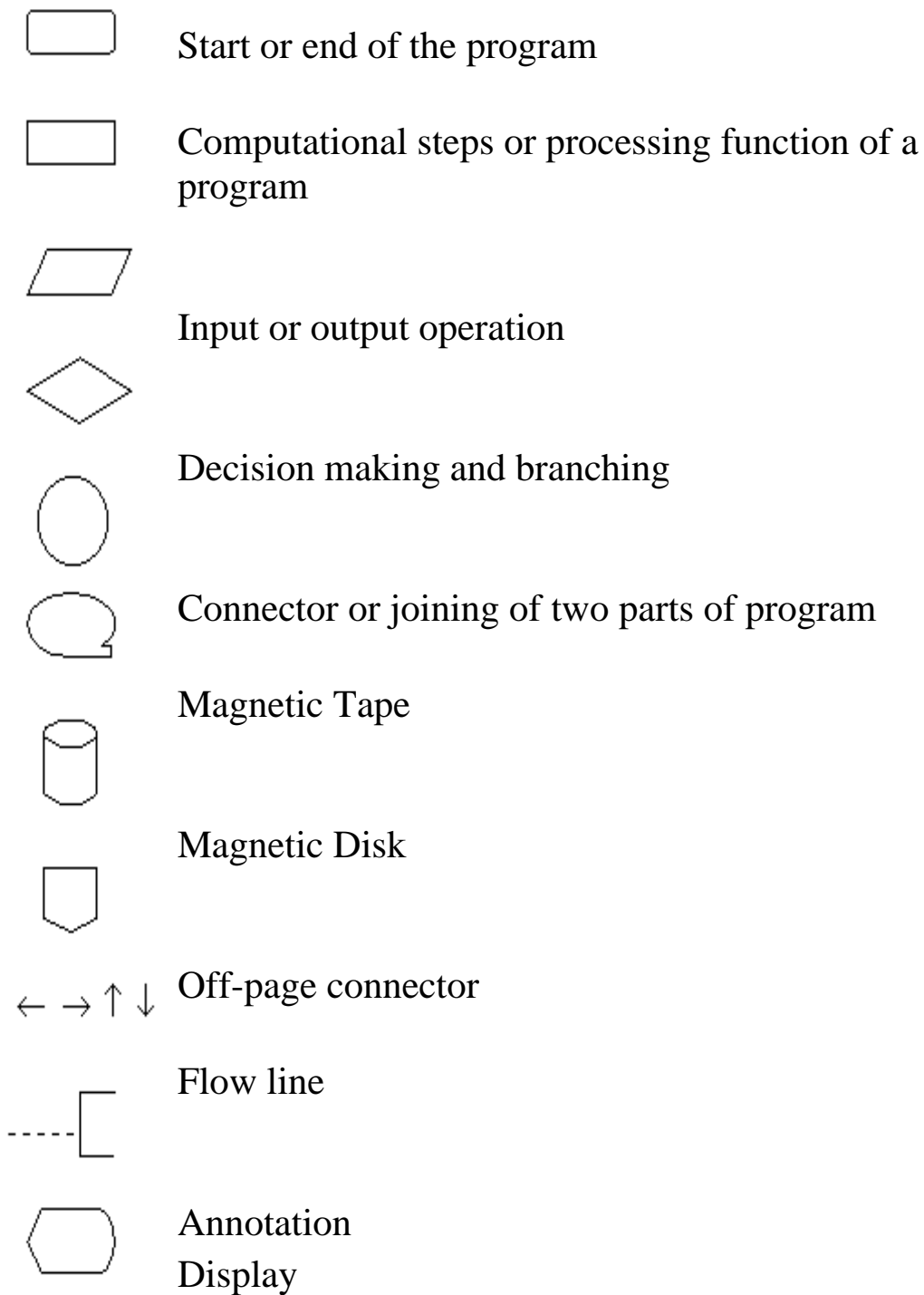
|	
▭	Start or end of the program
▭	Computational steps or processing function of a program
▱	Input or output operation
◇	Decision making and branching
○	Connector or joining of two parts of program
○	Magnetic Tape
⬡	Magnetic Disk
⬠	Off-page connector
← → ↑ ↓	Flow line
⌐	Annotation
⬡	Display

Fig. 25.1 Flowchart Symbols

The following are some guidelines in flowcharting:

a. In drawing a proper flowchart, all necessary requirements should be listed out in logical order.
b. The flowchart should be clear, neat and easy to follow. There should not be any room for ambiguity in understanding the flowchart.
c. The usual direction of the flow of a procedure or system is from left to right or top to bottom.
d. Only one flow line should come out from a process symbol.

or
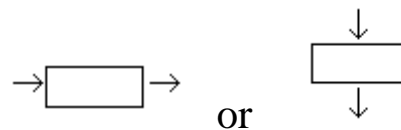
e. Only one flow line should enter a decision symbol, but two or three flow lines, one for each possible answer, should leave the decision symbol.
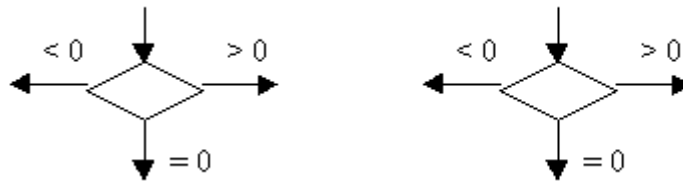
f. Only one flow line is used in conjunction with terminal symbol.

g. Write within standard symbols briefly. As necessary, you can use the annotation symbol to describe data or computational steps more clearly.

This is top secret data
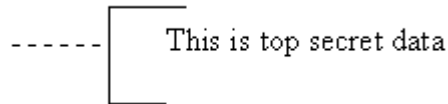
h. If the flowchart becomes complex, it is better to use connector symbols to reduce the number of flow lines. Avoid the intersection of flow lines if you want to make it more effective and better way of communication.
i. Ensure that the flowchart has a logical *start* and *finish*.
j. It is useful to test the validity of the flowchart by passing through it with a simple test data.

25.5 ADVANTAGES OF USING FLOWCHARTS

The benefits of flowcharts are as follows:

1. Communication: Flowcharts are better way of communicating the logic of a system to all concerned.
2. Effective analysis: With the help of flowchart, problem can be analysed in more effective way.
3. Proper documentation: Program flowcharts serve as a good program documentation, which is needed for various purposes.
4. Efficient Coding: The flowcharts act as a guide or blueprint during the systems analysis and program development phase.
5. Proper Debugging: The flowchart helps in debugging process.
6. Efficient Program Maintenance: The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part

LIMITATIONS OF USING FLOWCHARTS

1. Complex logic: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
2. Alterations and Modifications: If alterations are required the flowchart may require re-drawing completely.
3. Reproduction: As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.
4. The essentials of what is done can easily be lost in the technical details of how it is done.

Example:
-Write a program that reads a number and prints out "even" if the number is even or "odd" if the number is odd.
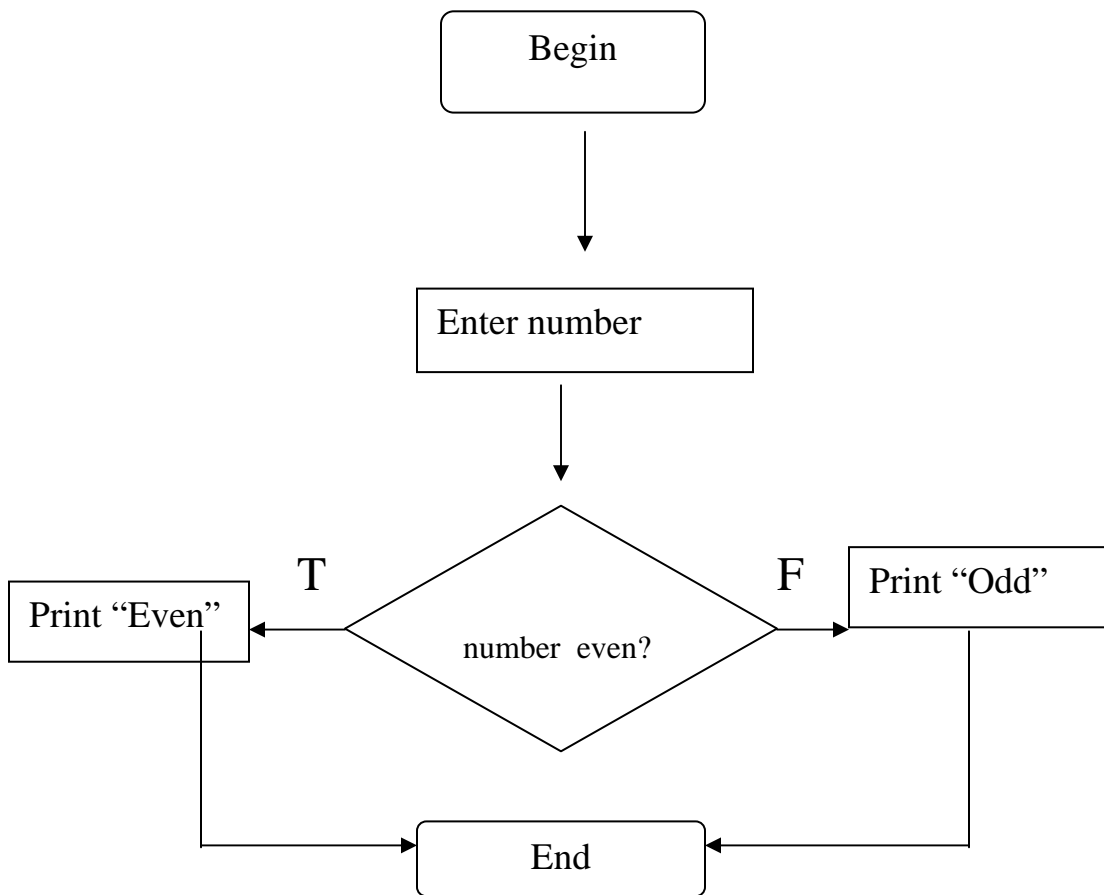
<u>Pseudocode:</u>
Input number
If number is even
    Then Print "even"
Else print "odd"

Flowchart:

```
              ┌─────────────┐
              │    Begin    │
              └─────────────┘
                     │
                     ▼
              ┌─────────────┐
              │ Enter number│
              └─────────────┘
                     │
                     ▼
┌──────────────┐  T      ╱◇╲       F  ┌─────────────┐
│ Print "Even" │◄───── number even? ─────►│ Print "Odd" │
└──────────────┘         ╲◇╱           └─────────────┘
        │                                      │
        └──────────►┌─────────────┐◄───────────┘
                    │     End     │
                    └─────────────┘
```

Problem:
Draw the flowchart for the following problem:
Write a program that prompts the user to enter a number and writes:
- "A negative number has been entered" if the number is negative.
- "0 has been entered" if the number is 0
- "A positive number has been entered" if the number is positive.