Unit – 1: Flowchart and Algorithm

- **Flowchart**
  - **Definition and Importance of flowchart**
  - **Symbols of Flowchart.**
  - **Flow lines, Terminals, Input/Output ,Processing Decision, Connection off page connectors**
  - **Guidelines for preparing Flowchart.**
  - **Flowchart structure**
    - **Sequence, selection, repetition.**
  - **Limitation of flowchart**
  - **Algorithm**
    - **Developing and writing algorithm using pseudo codes**

QUESTION:

1. Define flow chart and algorithm. (2 marks)
2. Write difference between flow chart and algorithm. (4 marks)
3. List out advantages and disadvantages of flow chart. (2 marks)
4. Draw flow chart to find square of a number. (3 marks)
5. Write algorithm to calculate area of a circle. (3 marks)
6. Draw symbols of flow chart and explain use of each. (4 marks)
7. Discuss structure of flow chart. (4 marks)

INTRODUCTION

- Intelligence is one of the key characteristics which differentiate a human being from other living creatures on the earth.
- Basic intelligence covers day to day problem solving and making strategies to handle different situations which keep arising in day to day life.
- One person goes Bank to withdraw money. After knowing the balance in his account, he/she decides to with draw the entire amount from his account but he/she has to leave minimum balance in his account. Here deciding about how much amount he/she may with draw from the account is one of the examples of the basic intelligence.
- During the process of solving any problem, one tries to find the necessary steps to be taken in a sequence. In this Unit you will develop your understanding about problem solving and approaches.

PROBLEM SOLVING

- Can you think of a day in your life which goes without problem solving? Answer to this question is of course, No. In our life we are bound to solve problems. In our day to day activity such as purchasing something from a general store and making payments, depositing fee in college, or withdrawing money from bank account.
- All these activities involve some kind of problem solving. It can be said that whatever activity a human being or machine do for achieving a specified objective comes under problem solving. To make it clearer, let us see some other examples.
- **Example1**: If you are watching a news channel on your TV and you want to change it to a sports channel, you need to do something i.e. move to that channel by pressing that channel number on your remote. This is a kind of problem solving.
- **Example 2**: If someone asks to you, what is time now? So seeing time in your watch and telling him is also a kind of problem solving.
- Now, broadly we can say that problem is a kind of barrier to achieve something and problem solving is a process to get that barrier removed by performing some sequence of activities (steps).
- If you can solve a given problem then you can also write an algorithm for it. In next section we will learn what an algorithm is.

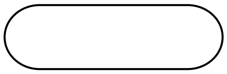
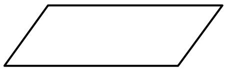
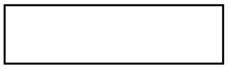
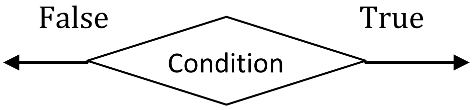
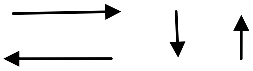
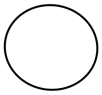
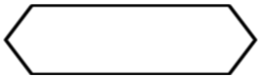
Flowchart

Definition and Importance of flowchart

- **Flowchart** is graphical or diagrammatical representation of sequence of any problem/program to be solved by programming language.
- It is used to prepare first before to write any program.
- Through the flowchart it is easy to understand the Logic and Sequence of problem.
- **Advantages of flowchart:**
  - The Flowchart is an excellent way of communicating the logic of a program.
  - It is easy and efficient to analyze problem using flowchart.
  - During program development cycle, the flowchart plays the role of a guide or a blueprint. Which makes program development process easier?
  - After successful development of a program, it needs continuous timely maintenance during the course of its operation. The flowchart makes program or system maintenance easier.
  - It helps the programmer to write the program code.
  - It is easy to convert the flowchart into any programming language code as it does not use any specific programming language concept.

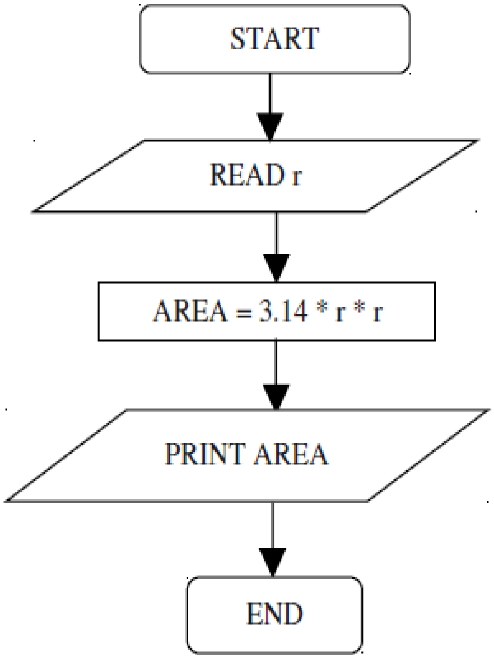
- **Disadvantage of flowchart**
  - The flowchart can be complex when the logic of a program is quite complicated.
  - Drawing flowchart is a time-consuming task.
  - Difficult to alter the flowchart. Sometimes, the designer needs to redraw the complete flowchart to change the logic of the flowchart or to alter the flowchart.
  - Since it uses special sets of symbols for every action, it is quite a tedious task to develop a flowchart as it requires special tools to draw the necessary symbols.
  - In the case of a complex flowchart, other programmers might have a difficult time understanding the logic and process of the flowchart.
  - It is just a visualization of a program, it cannot function like an actual program.

- **Symbols of Flowchart**.
  - **Flowchart** is graphical or diagrammatical representation of sequence of any problem/program to be solved by programming language.
  - It is used to prepare first before to write any program.
  - Through the flowchart it is easy to understand the Logic and Sequence of problem.
  - There are 7 basic symbols commonly used in flowcharting of any problem/program: **Terminal, Process, input/output, Decision, Connector and Predefined Process.** This is not a complete list of all the possible flowcharting symbols; it is the ones used most often.
  - **Various symbols are used to represent flowchart**

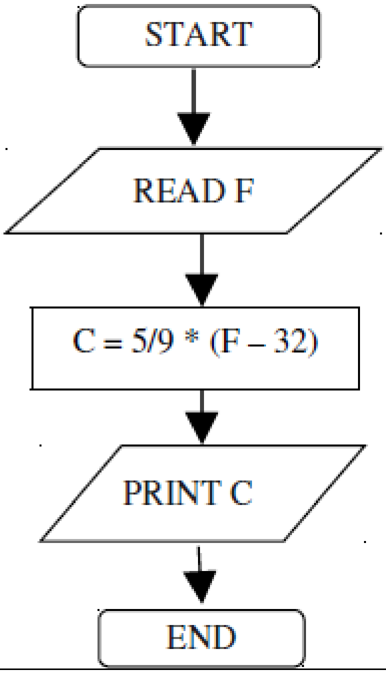
Start and Stop(Terminal) : This symbol is used to start and stop or end flow chart.	
Input and Output : This symbol is used for input and output operation.	
Process : This symbol is used to represent any assignment or expression.	
Decision: This symbol is used for any decision making statements.	
Direction : This symbol indicates flow of sequence or data	
Connector: This symbol is used for connecting flow lines from different places.	
Loop: This symbol is used for iteration or looping statements	

- **Example: Flowchart**

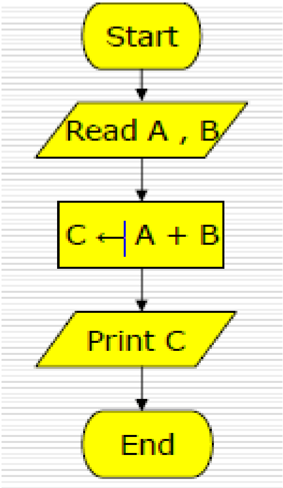
Problem1: Find the area of a circle of radius r.



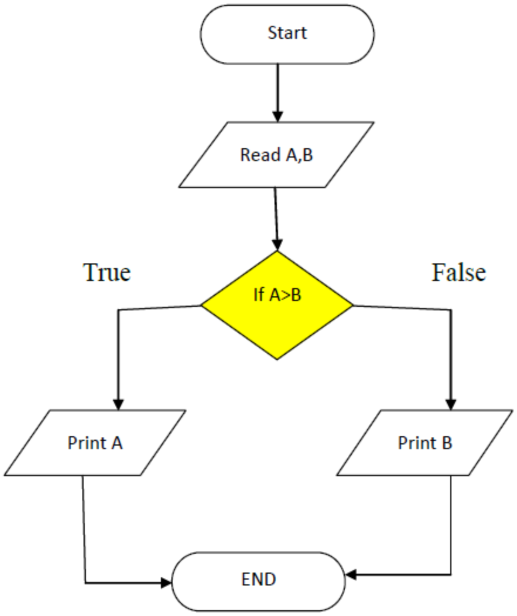
Problem 2: Convert temperature Fahrenheit to Celsius.



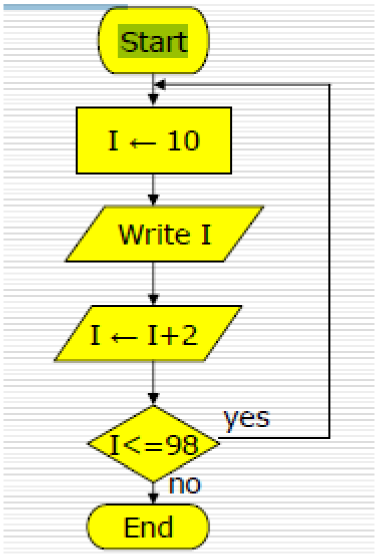
Problem3: Flowchart for an algorithm which gets two numbers and prints sum of their value



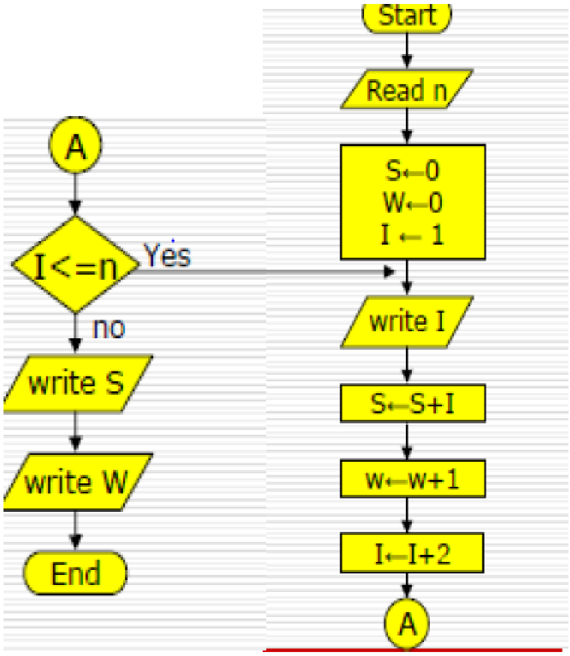
Problem5: Algorithm for find the greater number between two numbers.



Problem6*: Flowchart for the problem of printing even numbers between 9 and 100:*



Problem7: Flowchart for the problem of printing odd numbers less than a given number. It should also calculate their sum and count.



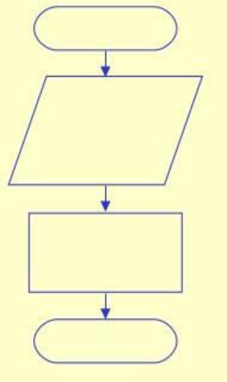
❖ **Guidelines for preparing Flowchart**.

- All boxes of the flowchart are connected with Arrows. (Not lines)
- Flowchart symbols have an entry point on the top of the symbol with no other entry points. The exit point for all flowchart symbols is on the bottom except for the Decision symbol.
- The Decision symbol has two exit points; these can be on the sides or the bottom and one side.
- Generally a flowchart will flow from top to bottom. However, an upward flow can be shown as long as it does not exceed 3 symbols.
- Connectors are used to connect breaks in the flowchart. Examples are:
  - From one page to another page.
  - From the bottom of the page to the top of the same page.
  - An upward flow of more than 3 symbols
- Subroutines and Interrupt programs have their own and independent flowcharts.
- All flow charts start with a Terminal or Predefined Process (for interrupt programs or subroutines) symbol.
- All flowcharts end with a terminal or a contentious loop.
- Flowcharting uses symbols that have been in use for a number of years to represent the type of operations and/or processes being performed.
- The standardized format provides a common method for people to visualize problems together in the same manner. The use of standardized symbols makes the flow charts easier to interpret; however, standardizing symbols is not as important as the sequence of activities that make up the process.

Flowchart structure

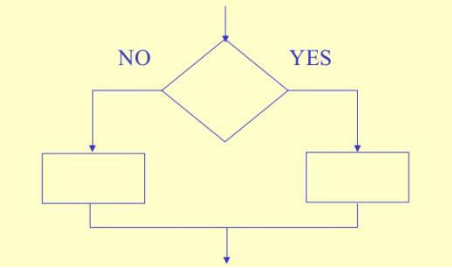
Sequence

- A series of actions are preformed in sequence
- The pay calculating example was sequence flowchart



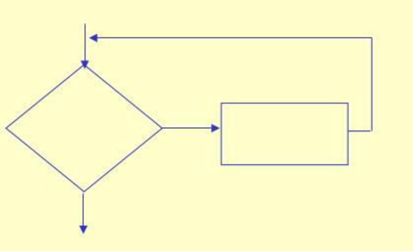
Selection

- One of two condition in taken, depending on a condition/selection
- A new symbol, the diamond, indicate yes/no question. If the answer to the question is yes, the flow follows one path. If the answer is no, the flow follows another path



Repetition

- A repetition structure represents part of the program that repeats. This type of structure is commonly known as a loop
- Notice the use of the diamond symbol. A loop tests a condition, and if the condtion exists, it performs an action, then it tests the condtion again. If the condtion still exists, the actio is repeated. This continues until the condtion no longer exists.



Limitation of flowchart

- Complex Logic: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex. In such cases, flowcharts tend to be over simplifying a process.
- Alterations and Modifications: If alterations are required the flowchart may require re-drawing completely. This is quite time and effort-consuming.
- Reproduction: As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.
- The essentials of what is done can easily be lost in the technical details of how it is done.
- Factors that affect the sequence are not included.

❖ Algorithm

- Algorithm can be defined as: “A stepwise representation of given problem.”
- “A sequence of activities to be processed for getting desired output from a given input.”
- Another definition of an algorithm as: “A formula or set of steps for solving a particular problem.
- To be an algorithm, a set of rules must be unambiguous and have a clear stopping point”. There may be more than one way to solve a problem, so there may be more than one algorithm for a problem.
- Now, if we take definition of algorithm as: “A sequence of activities to be processed for getting desired output from a given input.” Then we can say that:
  1. Getting specified output is essential after algorithm is executed.
  2. One will get output only if algorithm stops after finite time.
  3. Activities in an algorithm to be clearly defined in other words for it to be unambiguous.
- Before writing an algorithm for a problem, one should find out what is/are the inputs to the algorithm and what is/are expected output after running the algorithm. Now let us take some exercises to develop an algorithm for some simple problems: While writing algorithms we will use following symbol for different operations:

  ‘+’ for Addition
  ‘-’ for Subtraction
  ‘*’ for Multiplication
  ‘/’ for Division and
  ‘← ’ for assignment. For example A← X*3 means **A** will have a value of **X*3.**
- Type of Algorithms (Que:Discuss structure of flow chart/algorithm)
  - The *algorithm and flowchart*, classification to the three types of **control structures**. They are:
    - Sequence
    - Branching (Selection)
    - Loop (Repetition)
- **These three control structures are sufficient for all purposes**.
  - **The sequence** is exemplified by sequence of statements place one after the other – the one above or before another gets executed first. In flowcharts, sequence of statements is usually contained in the rectangular process box.
  - **Example of Algorithm**
  - **Problem 1: Find the area of a Circle of radius r.**
    - Algorithm:

```
Step1: Read\input r
Step2: Area← 3.14*r*r // calculation of area
Step3: Print Area
```

  - **Problem2: Write an algorithm to read two numbers and find their sum.**
    - **Algorithm:**

```
Step1: Start
Step2: Read\input the first num1.
Step3: Read\input the second num2.
Step4: Sum← num1+num2 // calculation of sum
Step5: Print Sum
Step6: End
```

- **The branch** refers to a binary decision based on some condition. If the condition is true, one of the two branches is explored; if the condition is false, the other alternative is taken. This is usually represented by the 'if-then' construct in pseudo-codes and programs. In flowcharts, this is represented by the diamond-shaped decision box. This structure is also known as the selection structure.

  - **Problem1: write algorithm to find the greater number between two numbers**

```
Step1: Start
Step2: Read/input A and B
Step3: If A > B then C=A
Step4: if B > A then C=B
Step5: Print C
Step6: End
```

  - **Problem2: write algorithm to find the** $f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$ **result of equation:**

```
Step1: Start
Step2: Read/input x
Step3: If X < 0 then F=-X
Step4: if X >= 0 then F=X
Step5: Print F
Step6: End
```

  - **Problem3: A algorithm to find the largest value of any three numbers.**

```
Step1: Start
Step2: Read/input A,B and C
Step3: If (A>=B) and (A>=C) then Max=A
Step4: If (B>=A) and (B>=C) then Max=B
Step5:If (C>=A) and (C>=B) then Max=C
Step6: Print Max
Step7: End
```

- **The loop** allows a statement or a sequence of statements to be repeatedly executed based on some loop condition. It is represented by the 'while' and 'for' constructs in most programming languages, for unbounded loops and bounded loops respectively. (Unbounded loops refer to those whose number of iterations depends on the eventuality that the termination condition is satisfied; bounded loops refer to those whose number of iterations is known before-hand.)
- In the flowcharts, a back arrow hints the presence of a loop. A trip around the loop is known as iteration. You must ensure that the condition for the termination of the looping must be satisfied after some finite number of iterations, otherwise it ends up as an infinite loop, a common mistake made by inexperienced programmers. The loop is also known as the repetition structure.
- **Examples:**
  - **Problem1: An algorithm to calculate even numbers between 0 and 99**

```
1. Start
2. I ← 0
3. Print I
4. I ← I+2
5. If (I <=98) then go to line 3
```

```
6. End
```

- **Problem2: Design an algorithm which gets a natural value, n as its input and calculates odd numbers equal or less than n. Then write them in the standard output:**

```
1. Start
2. Read n
3. I ← 1
4. Write I
5. I ← I + 2
6. If (I <= n) then go to line 4
7. End
```

- **Problem3: Design an algorithm which generates even numbers between 1000 and 2000 and then prints them in the standard output. It should also print total sum:**

```
1. Start
2. I ← 1000 and Sum ← 0
3. Write I
4. Sum ← Sum + I
5. I ← I + 2
6. If (I <= 2000) then go to line 3
else go to line 7
7. Write S
8. End
```

- **Problem4: Design an algorithm with a natural number, n, as its input which calculates the following formula and writes the result in the standard output:**

```
S = ½ + ¼ + … +1/n
1. Start
2. Read n
3. I ← 2 and S ← 0
4. S= S + 1/I
5. I ← I + 2
6. If (I <= n) then go to line 4
else write S in standard output
7. End
```

❖ Properties of algorithm

- ➢ Donald Ervin Knuth has given a list of five properties for algorithm, these properties are:
  - **Finiteness:** An algorithm must always terminate after a finite number of steps. It means after every step one reach closer to solution of the problem and after a finite number of steps algorithm reaches to an end point.
  - **Definiteness:** Each step of an algorithm must be precisely defined. It is done by well thought actions to be performed at each step of the algorithm. Also the actions are defined unambiguously for each activity in the algorithm.
  - **Input:** Any operation you perform need some beginning value/quantities associated with different activities in the operation. So the value/quantities are given to the algorithm before it begins.

- Output: One always expects output/result (expected value/quantities) in terms of output from an algorithm. The result may be obtained at different stages of the algorithm. If some result is from the intermediate stage of the operation then it is known as intermediate result and result obtained at the end of algorithm is known as end result. The output is expected value/quantities always have a specified relation to the inputs.
- Effectiveness: Algorithms to be developed/written using basic operations. Actually operations should be basic, so that even they can in principle be done exactly and in a finite amount of time by a person, by using paper and pencil only.

❖ **Advantages of algorithm**

- It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
- An algorithm uses a definite procedure.
- It is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.
- Every step in an algorithm has its own logical sequence so it is easy to debug.
- By using algorithm, the problem is broken down into smaller pieces or steps hence, it is easier for programmer to convert it into an actual program

❖ **Disadvantages of algorithm.**

- Writing algorithm takes a long time.
- An Algorithm is not a computer program; it is rather a concept of how a program should be.
- Here, it is very difficult to show looping and braching.

❖ **Que:-Give the difference between Algorithm and Flowchart.**

Algorithm	**Flow Chart**
An algorithm is the finite set of step-by-step instructions that solve a program.	A flow chart is a graphical or diagrammatic representation of a problem.
An algorithm gives verbal representation	Flowchart is graphical representation
Algorithm gives verbal representation which is almost similar to English language. Suitable for large and modular programs.	Gives pictorial representation Suitable for small programs.
Easier to understand	To understand flowchart one has to be familiar with symbols.
Drawing tools are not required	Drawing tools are required.