Unit II – Part 01
(Solution Formulation)

Lecture Slide

AS2023

Objectives

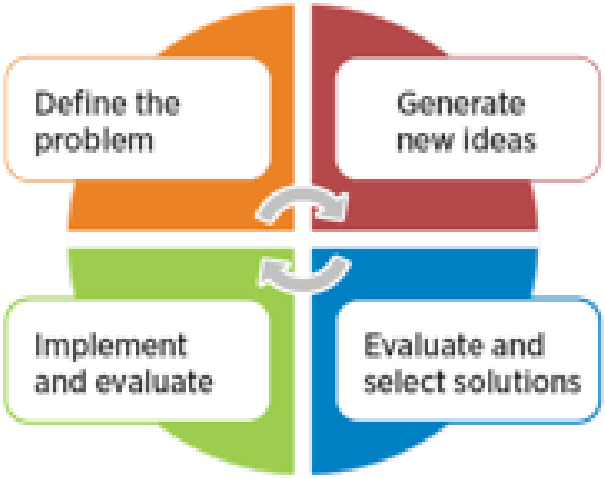
By the end of this session, students will be able to:

- Define problem
- Explain top-down approach
- Define algorithm
- Explain Flowchart & Pseudocode
- Differentiate between various selection methods
- Differentiate between two looping methods
- Benefits of flowchart and pseudocode

Problem

Problem Solving

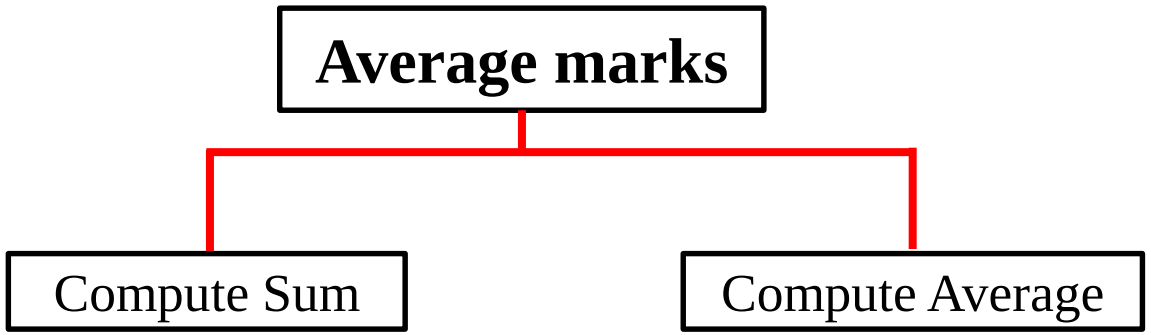
Process of finding an efficient solution to a given problem.



- Requires careful thought process and planning
- Computer lacks natural intelligence like us

Top-down Approach

- A given problem is successively broken down into smaller and smaller sub-problems or operations
- Sub-problem should be easily solvable
- Consider a problem to compute average marks of a student



Algorithm

- How do you prepare tea?
- Consider the following steps:
  1. If the kettle doesn't contain water, then fill it
  2. Plug the kettle into the power point and switch it on
  3. If the teapot is not empty, then empty the tea pot
  4. Place the tea leaves in the pot
  5. If the water in the kettle is not boiling, then go to step 2
  6. Switch off the kettle
  7. Pour water from the kettle into tea pot

Algorithm (Cont..)

- The step-wise step process of preparing represents the Algorithm
- **Define Algorithm.**
  - Effective procedure for solving a problem in a finite number of steps

  (or)

  - Finite number of sequential steps to solve a problem
- Algorithm can be represented using **Flowchart** and **Pseudocode**

Key features of Algorithm

Algorithm Features:

1. Sequence (known as process)
2. Decision or selection
3. Repetition/iteration/looping

1. If the kettle doesn't contain water,
   then fill it
2. Plug the kettle into the power point and switch it on
3. If the teapot is not empty, then empty the tea pot
4. Place the tea leaves in the pot
5. If the water in the kettle is not boiling, then go to step 2
6. Switch off the kettle
7. Pour water from the kettle into tea pot

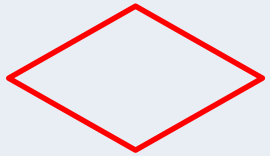
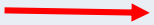
Characteristics

- **Precision** – the steps are precisely stated(defined).
- **Uniqueness** – results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
- **Finiteness** – the algorithm stops after a finite number of instructions are executed.
- **Input** – the algorithm receives input.
- **Output** – the algorithm produces output.
- **Generality** – the algorithm applies to a set of inputs

Flowchart

- Pictorial representation of algorithms
- Steps are drawn in the form of different shapes
- Logical flow is indicated by interconnecting arrows
- Represents the information flow in the program

Flowchart Symbols

Symbol	Name	Description
	Processing	Used for representing arithmetic and data movement instructions.
	Decision	Denotes decision (branch) to be made. Program should continue along one of the two routes (IF/ELSE). The symbol has one entry and two exit paths. The path chosen depends on whether the answer to question is yes or no
	Input /output	Represents information entering or leaving the systems such as customer order (input) and service (output)
	Flow lines	Connect symbols. And indicates the sequence of steps and direction of flow of control
	Terminal	Used to represent beginning (start), the termination (end or stop) or halt (pause) in the program logic
	connector	Used to join different flow lines

Guidelines for Flowchart

- Following guidelines are used for create flowchart
  - The flowchart should be neat, clear and easy to follow
  - The flow chart must have logical start and finish
  - In drawing a proper flowchart, all necessary requirements should be listed in logical order
  - Only one flow line should come out from a process symbol
  - Only one flow line should enter a decision symbol, however two or three flow lines (one for each possible answer) may leave the decision symbol

Guidelines (Cont..)

- Following guidelines are used
  - Only one flow line is used with terminal symbol
  - Write briefly with standard symbols. If necessary , use annotation symbol to describe data or process more clearly
  - Incase of complex flowchart, connectors symbols are used to reduce the number of flow lines
  - Intersection of flow lines should be avoided to make it more effective and better way of representing communication

Benefits of Flowchart

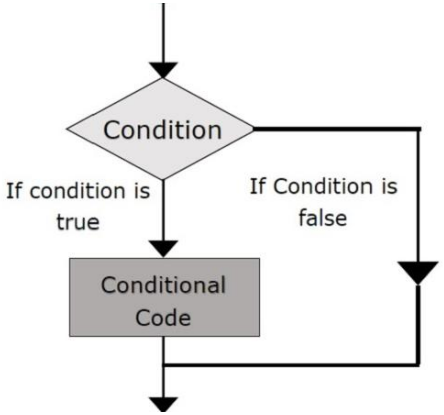
- The following are the benefits of flowchart
  - Makes logic clear
  - Communication
  - Effective analysis
  - Useful in coding
  - Proper testing and debugging
  - Appropriate documentation

Limitations of Flowchart

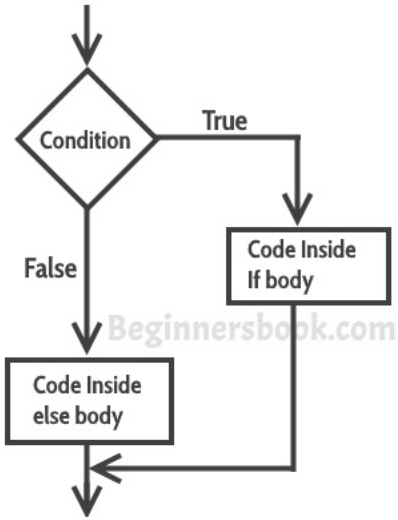
- The following are the limitations of flowchart
  - Complex
  - Costly
  - Difficult to modify
  - No update

Concept of Selection

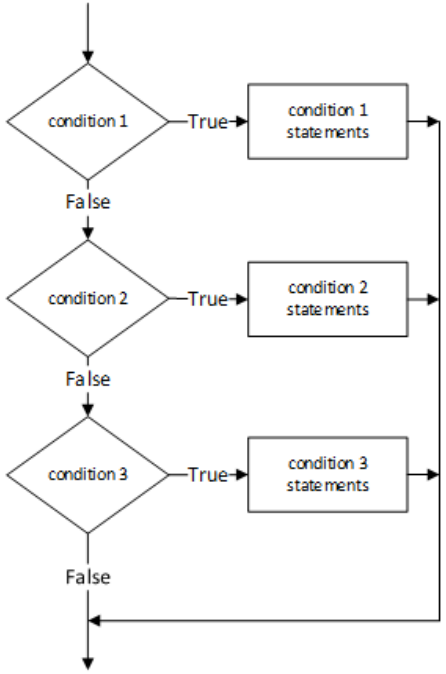
- Allows to make decision by testing several conditions



if statement



if else statement

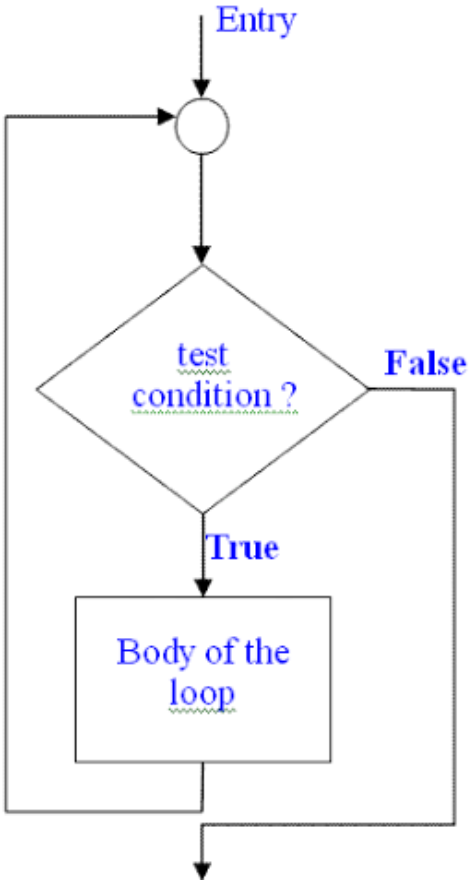


if else if statement

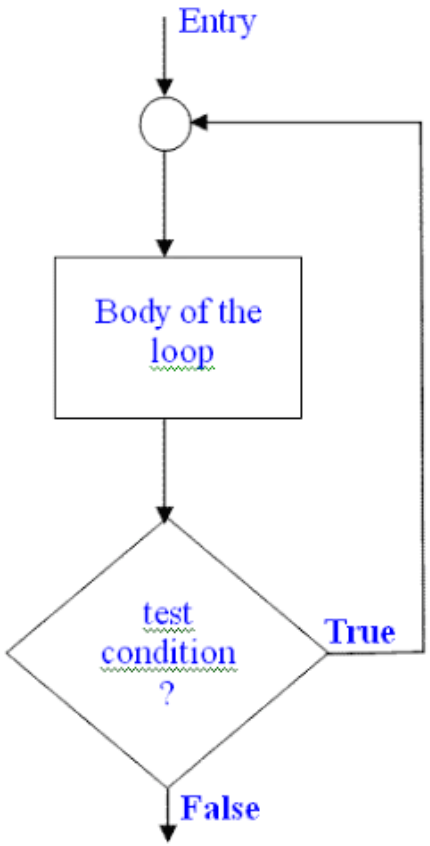
Iteration or loop

- Directs the system to loop back to a previous statement in the program, repeating the same sequence over and again, usually with new data
- When a sequence of statement is repeated against, a condition is said to be in loop
- Using looping, the programmer avoids writing the same set of instruction again
- The looping process can either be one time or multiple times until the desired output is obtained within a single program

Iteration or loop (cont..)



Entry controlled Loop



Exit controlled Loop

Pseudocode

- Pronounced as *soo-doh-kohd*
- ***Pseudo*** means imitation & ***code*** refers to instruction in programming languages
- It is a generic way of describing the algorithm without using any specific programming language-related notations
- Or pseudocode uses plain English rather than using symbols
- Also known as Program Design Language (PDL)

Pseudocode (Cont..)

- Pseudocode uses some keywords to denote programming process
  - **Input:** READ, OBTAIN, GET & PROMPT
  - **Output:** PRINT, DISPLAY, and SHOW
  - **Compute:** COMPUTE, CALCULATE, and DETERMINE
  - **Initialize:** SET, and INITIALIZE
  - **Adding/Increasing counter:** INCREMENT
  - **Subtracting/Decreasing counter:** DECREMENT

General guidelines for Pseudocode

- Statements should be written in simple English and should be programming language independent
- Steps must be understandable
- Pseudocode should be concise
- Each instruction should be written in separate line and each statement should express just one action
- Capitalize keywords like PRINT, READ and so on
- It should allow easy transition from design to coding in programming language

Benefits of Pseudocode

- Allows developer to express the design in plain natural language
- Easier to develop to program from a pseudocode than with a flowchart
- Unlike flowchart, pseudocode is compact and doesn't tend to run over many pages. Its simple structure and readability makes it easier to modify

Limitations of Pseudocode

- Does not provide visual representation
- No accepted standards for writing pseudocode

Selection statement

```
IF (<condition>):
    THEN <statement>;
ENDIF;
```

if statement

```
IF (<condition>):
    THEN <statement0>;
    ELSE <statement1>;
ENDIF;
```

if else statement

```
IF (<condition>):
    THEN <statement0>;
     ELSE IF (<condition>):
         THEN <statement0>;
         ELSE <statement1>;
           ---
           ---
ENDIF;
```

If else if statement

Iteration Statement

```
<statement0>;
WHILE <Condition>:
    THEN <statement1>;
    <statement2>;
ENDWHILE;
```

OR

```
FOR (<INITIALIZE var>
<Condition>):
   THEN <statement1>;
   <statement2>;
ENDFOR;
```

Entry Controlled Loop

```
DO:
  <statement0>;
WHILE <Condition>;
ENDDOWHILE;
```

Exit Controlled Loop

Thank you