

Limitations, Control Flow, and Pseudocode

Unit 1: Problem Solving using Flowchart and Algorithm

Lecture 5: Limitations of flowchart, Flow of control, Pseudocode

Agenda

Limitations of Flowcharts

Concept of Control Flow

Types of Control Flow

What is Pseudocode?

Advantages of Pseudocode

Writing Pseudocode Examples

Summary of Unit 1

Limitations of Flowcharts

Complex Logic: Become incredibly messy and unreadable for large, complex programs.

Modifications: Difficult to alter. Often require redrawing the entire flowchart.

Reproduction: Hard to type out; requires specific software tools to draw cleanly.

Lack of standard detailing: No standard way to express detailed data structures.

Flow of Control

Flow of control refers to the order in which individual statements, instructions, or function calls are executed.

It determines how the program progresses from start to finish.

Without control flow, a program executes strictly top-to-bottom.

Types of Control Flow

1. Sequential: Default mode. Line 1, then Line 2, etc.
2. Selection (Branching): Making decisions (If/Else). Paths diverge.
3. Iteration (Looping): Repeating a block of code multiple times (For/While).

What is Pseudocode?

Pseudocode ('fake code') is an informal, high-level description of an algorithm.

Uses structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

No strict syntax rules.

Why use Pseudocode?

Solves flowchart limitations: Easy to write in any text editor, easy to modify.

Scales better for complex logic.

Focuses on logic without getting bogged down by syntax errors.

Easily translated into actual programming code (like Python).

Basic Pseudocode Keywords

Input/Output: READ, GET, PRINT, DISPLAY

Processing: COMPUTE, CALCULATE, SET, INITIALIZE

Selection: IF, THEN, ELSE, ENDIF

Iteration: WHILE, DO, FOR, ENDWHILE

Example: Factorial of a Number

```
READ n
SET fact = 1
SET i = 1
WHILE i <= n DO
    fact = fact * i
    i = i + 1
ENDWHILE
PRINT fact
```

Example: Largest of Three (Pseudocode)

```
READ a, b, c
IF a > b AND a > c THEN
    PRINT a
ELSE IF b > a AND b > c THEN
    PRINT b
ELSE
    PRINT c
ENDIF
```

Comparing the Tools

Plain English: Vague, ambiguous.

Flowchart: Great visual, hard to maintain/scale.

Pseudocode: Best balance for complex logic, easy to edit.

Python Code: Final step, requires strict syntax.

Unit 1 Summary

Problem solving is a systematic process.

Algorithms are step-by-step logical solutions.

Flowcharts visualize control flow.

Pseudocode bridges the gap between logic and implementation.

You are now ready to start coding in Python!