📘 Understanding Pseudocode and Translating It to C++

🧠 What is Pseudocode?

Pseudocode is a high-level, informal way of describing an algorithm using plain language and programming-like structure. It is not bound by the syntax rules of any specific programming language, making it ideal for planning and communicating logic before implementation.

🔍 Key Characteristics:

- Language-agnostic
- Focuses on logic, not syntax
- Easy to read and write
- Often used in algorithm design and documentation

🧱 Structure of Pseudocode

Pseudocode typically includes:

- **Control structures**: `IF`, `ELSE`, `WHILE`, `FOR`
- **Function definitions**: `PROCEDURE`, `FUNCTION`
- **Input/Output**: `READ`, `PRINT`
- **Comments**: Often written with `//` or `#`

Example:

```
FUNCTION Factorial(n)
    IF n == 0 THEN
        RETURN 1
    ELSE
        RETURN n * Factorial(n - 1)
    ENDIF
END FUNCTION
```

🔄 Translating Pseudocode to C++

To convert pseudocode into C++, follow these steps:

1. Identify the structure

- Functions → `returnType functionName(parameters)`
- Conditionals → `if`, `else`
- Loops → `for`, `while`
- I/O → `cin`, `cout`

2. Add C++ syntax

- Use semicolons, braces `{}`, and proper types (`int`, `double`, etc.)
- Include headers like `#include <iostream>`

🧮 Example: Translating Factorial

🧾 Pseudocode:

```
FUNCTION Factorial(n)
    IF n == 0 THEN
        RETURN 1
    ELSE
        RETURN n * Factorial(n - 1)
    ENDIF
END FUNCTION
```

💻 C++ Code:

```cpp
#include <iostream>
using namespace std;

int Factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * Factorial(n - 1);
}

int main() {
    int num = 5;
    cout << "Factorial of " << num << " is " << Factorial(num) << endl;
    return 0;
}
```

🏏 Practice Example

🧾 Pseudocode:

```
READ n
sum ← 0
FOR i ← 1 TO n DO
    sum ← sum + i
ENDFOR
PRINT sum
```

💻 C++ Code:

```cpp
#include <iostream>
using namespace std;

int main() {
    int n, sum = 0;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        sum += i;
    }
    cout << "Sum is " << sum << endl;
    return 0;
}
```

🧠 Translating Tips

- go line-by-line. Don't get overwhelmed by the whole method.
- Don't change variable names until you have a working version.
- Often times, you'll have to add to it to make it fit in your application.
- := means assignment.
- Don't change the logic of the pseudocode.
- Indention means blocks and indicates what block the code belongs to.
- It's ok to skip something and come back to it later.
- end if, end for, end procedure means `}` in C++
- If it doesn't work after translating, go back and compare your code line-by-line (sometimes character by character) with the pseudocode. Sometimes the problem is simple translation mistake.

✅ Summary

Pseudocode Element	C++ Equivalent
`READ x`	`cin >> x;`
`PRINT x`	`cout << x;`
`IF`, `ELSE`	`if`, `else`
`FOR`, `WHILE`	`for`, `while`
`EndIF`, `EndFor`	`}`
`EndWhile`	`}`
`EndProcedure`	`}`
`FUNCTION`	`returnType functionName()`

🎯 Quiz!

Here's a short quiz on the topic: quiz

🔭 Markdown Viewer

How to view the markdown files in a browser...

- Markdown Viewer

🧠 Lecture Practices

Here are the lecture Practices...

- Day 4
- Day 5
- Day 6

🔍 Lecture Quizzes

Here are the lecture quizzes...

- Day 4
- Day 5
- Day 6

Weekly Topics

Here are the topics for the week...

- Recursion
- Pseudocode
- Sorting
- Searching
- Maps
- Time Complexity