Module 1g: Introduction to Problem Solving and Python Fundamentals

Premanand S

Assistant Professor,
School of Electronics and Engineering,
Vellore Institute of Technology, Chennai

premanand.s@vit.ac.in

July 19, 2025

What is Pseudocode?

- **Pseudocode** is a way to describe an algorithm using simple, human-readable language.
- It is not written in a specific programming syntax, but it resembles real code.
- It helps plan and visualize logic before writing actual code.
- No compiler or interpreter needed—just logic!

Why Use Pseudocode?

- **Clarifies Logic:** Focuses on the thought process without worrying about syntax.
- **Easy to Communicate:** Great for discussing ideas with team members or beginners.
- **Helps Debug Early:** Logical mistakes are easier to find before coding.
- **Bridge to Code:** Makes writing actual code easier—just translate the steps!
- **Documentation:** Acts as a blueprint or reference for the project.

Conditions in Pseudocode and Link to Programming

- Conditions use keywords like `IF`, `ELSE IF`, `ELSE`, and `END IF`.

Pseudocode

```
IF age $>= 18 THEN$
    PRINT ”Eligible to vote”
ELSE
    PRINT ”Not eligible”
END IF
```

Python Code

```python
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible")
```

- **Connection:** Pseudocode = logical sketch → Programming = implementation.

What NOT to Do in Pseudocode

- **Don't use actual programming syntax.**
  - Use plain English, not curly braces or language-specific keywords.
- **Don't focus on syntax errors.**
  - Pseudocode is for logic, not compiling.
- **Don't write full code or functions.**
  - No need to declare variables or imports—just describe the steps.
- **Avoid ambiguity.**
  - Be clear and structured—use consistent indenting and keywords.
- **Don't skip key decisions or steps.**
  - Every part of the logic should be visible, even if obvious.

Remember: Pseudocode is about *clarity*, not correctness.

Common constructs used in Pseudocode

- SEQUENCE: Represents linear tasks performed one after the other.
- IF-THEN-ELSE: Conditional statements that dictate different actions based on conditions.
- WHILE: Loops that continue as long as a condition is true.
- FOR: Loops that iterate a specific number of times.
- REPEAT-UNTIL: Loops that continue until a condition is met.
- CASE: A generalized form of IF-THEN-ELSE for multiple conditions.
- CALL: Used for invoking classes or calling functions.
- EXCEPTION: Used for handling exceptions, along with the WHEN keyword.

Pseudocode Example – Even or Odd

Problem

Check if a given number is even or odd.

Pseudocode

```
START
INPUT number
IF number MOD 2 = 0 THEN
    PRINT ”Even”
ELSE
    PRINT ”Odd”
END IF
END
```

Pseudocode Example – Largest of Three Numbers

Problem

Find the largest among three numbers.

Pseudocode

```
START
INPUT a, b, c
IF $a > b$ AND $a > c$ THEN
    PRINT ”a is largest”
ELSE IF $b > c$ THEN
    PRINT ”b is largest”
ELSE
    PRINT ”c is largest”
END IF
END
```

Pseudocode Example – Sum of N Natural Numbers

Problem

Find the sum of first N natural numbers.

Pseudocode

```
START
INPUT N
SET sum = 0
FOR i = 1 TO N DO
    sum = sum + i
END FOR
PRINT sum
END
```

Assignment

- Login Validation System – Check if username and password are correct.
- ATM Cash Withdrawal – Perform PIN check and balance verification.
- Electricity Bill Calculation – Compute bill based on slab-wise consumption.
- Even or Odd Number Checker – Determine if a number is even or odd.
- Largest of Three Numbers – Identify the greatest among three values.

Assignment

- Student Grade Calculator – Assign grades based on average marks.
- Simple Interest Calculator – Calculate SI using principal, rate, and time.
- Voting Eligibility Checker – Check if age is eligible for voting.
- Factorial of a Number – Use loop to compute factorial.
- Number Guessing Game – Loop until user guesses the correct number.

Pseudocode Assignment

- Temperature Converter – Convert Celsius to Fahrenheit.
- Leap Year Checker – Determine if a year is a leap year.
- Online Food Ordering – Select item, place order, and confirm payment.
- Sum of N Natural Numbers – Use loop to calculate the total.
- Traffic Light Controller – Simulate basic traffic signal changes.

Brushing Up

Revision 1: You can apply all the concepts

- Mr. Jones always gives True/False tests to his class. His tests always have 20 questions. The maximum class size is 35. He needs a program to calculate the students' grades based on the best score. Grade A will range from the best score, to the best score minus 2. B will range from the best score minus 3, to the best score minus 4. C will range from the best score minus 5, to the best score minus 6. D will range from the best score minus 7, to the best score minus 8. F will be anything below the best score minus 8. Each student's ID and test answers will be entered. The output will be each student's ID, number correct, and grade, along with the single highest score for the class. Develop a solution for Mr. Jones's problem. Use four one-dimensional arrays—one for the correct scores and the other three for the needed output.

Revision 2: You can apply all the concepts

- A restaurant manager wants to know how many employees are needed at the restaurant each hour of the day. The minimum number of employees needed at any hour is 3. After that, one additional employee is required for each 20 customers. The restaurant is open 24 hours a day. The manager has counted the number of customers each hour for 14 days. The manager will use the average number of customers for each hour over the 14 days to calculate the needed number of employees for each hour. Develop a solution to output the needed number of employees per hour. (There is no such thing as a partial employee.)

Revision 3: You can apply all the concepts

- An instructor has 30 students in her class. Each student is identified by a number from 1 to 30. Grades are stored in a one-dimensional array. The instructor would like to enter a student number and have the student's test score printed on the monitor. Develop a solution to output the needed information.

Verdict of Module 1

- Programming is a foundational digital skill that empowers us to build solutions, automate tasks, and innovate across industries.
- Python stands out as an ideal first language due to its simplicity, versatility, and widespread community support in domains like AI, data science, and web development.
- Structured problem-solving through PAC, algorithms, and flowcharts lays the groundwork for writing logical and effective programs.

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in
ring me: +91 73586 79961
Follow me: Linkedin
Medium Blogs
Analytics Vidhya: Blogs

Don’t just code — think, plan, and solve