

1) What is computational thinking?

Computational Thinking is a problem-solving method where we break problems into parts, spot patterns, focus on key details, and design step-by-step solutions like a computer.

2) What are the steps involved in computational thinking?

Steps in Computational Thinking are:

Decomposition – Break a big problem into smaller parts.

Pattern Recognition – Look for similarities or repeated ideas.

Abstraction – Focus only on the important details, ignore the rest.

Algorithm Design – Create clear step-by-step instructions to solve the problem.

3) What is an algorithm?

An algorithm is a clear, step-by-step set of instructions used to solve a problem or complete a task, in a proper order.

4) What are the different ways to represent an algorithm?

Ways to Represent an Algorithm are:

Flowchart – A diagram that uses shapes (like ovals, rectangles, diamonds, arrows) to show the steps and decisions in order.

Pseudocode – Writing the steps in simple, structured English that looks like code however is not related to any programming language.

5) What is a variable?

A variable is a named storage place in a computer's memory that holds data, which can change while the program runs.

6) Why do we use pseudocode?

We use pseudocode to plan and explain algorithms clearly before writing real code.

Key reasons are:

Easy to understand – written in plain English.

Focus on logic first – no need to worry about programming syntax.

Helps teamwork – programmers and non-programmers can follow the steps.

Reduces mistakes – makes coding smoother by clarifying the plan.

7) List some of the basic rules we follow while writing a pseudocode?

For Input

INPUT "Your Prompt for input" Variable_Name

For Variable assignment

Variable_Name <- data

For Output (simple string output)

OUTPUT "What you want to say"

For Output (simple string output with concatenation of variable)

OUTPUT "The sum of the numbers is", SUM

For Output (simple string concatenation)

OUTPUT "We are", "Grade 8", "ASTRA"

8) List some of the arithmetic operators used in pseudocode.

Some of the common arithmetic operators used in pseudocode are:

Addition (+) → adds numbers

Subtraction (−) → subtracts numbers

Multiplication (× or *) → multiplies numbers

Division (/) → divides numbers

Modulus (MOD or %)

9) List some of the comparison operators used on pseudocode?

Comparison Operators in Pseudocode are:

Equal to (=) → checks if two values are the same

Not equal to (≠) → checks if values are different

Greater than (>) → checks if one value is bigger

Less than (<) → checks if one value is smaller

Greater than or equal to (≥) → bigger or same

Less than or equal to (≤) → smaller or same

10) List some of the logical operators used in pseudocode?

Logical Operators in Pseudocode are:

AND → True only if both conditions are true

OR → True if at least one condition is true

NOT → Reverses the condition (True becomes False, False becomes True)

11) How do we follow the pseudocode?

Read step by step – go through each instruction in order.

Understand the logic – see what the algorithm is trying to do.

Execute mentally or on paper – apply the steps to given inputs.

Check decisions – follow the correct path when IF or ELSE appears.

Repeat loops – continue until the condition tells you to stop.

End properly – finish when the pseudocode says END.