

Section 1: Core Concepts

1. Connecting to Previous Learning

- **Algorithms:** A clear set of step-by-step instructions designed to solve a problem or complete a task.
- **Iteration (Loops):** Repeating instructions in code to avoid writing the same line multiple times.
- **Selection:** Using conditional statements (IF... THEN... ELSE) to choose which path or instructions to run based on whether a condition is True or False.

2. Key Concepts

A. Algorithm Efficiency

- **Definition:** An algorithm is efficient when it completes a task accurately by carrying out only necessary instructions without wasting time or computer power.
- **Making Algorithms Efficient:**
 1. **Using Iteration (Loops):** Grouping and repeating common instructions instead of listing each one individually.
 2. **Using Selection (IF... THEN... ELSE):** Checking conditions first so the computer carries out actions *only when needed*.

B. Decomposition

- **Definition:** Decomposition means breaking down a complex problem into smaller sub-problems that are easier to understand, plan, and solve.

C. The Four Building Blocks of Problem Planning

When planning computer solutions, break the problem into:

1. **Input:** Data or instructions given to the computer by a user (e.g., entering numbers on a keyboard, typing an answer to a question).
2. **Process:** Calculations, variable assignments, or data manipulations performed by the computer (e.g., calculating $5 + 5$, assigning a user's typed reply to a variable like `FavouriteAnimal`).
3. **Condition:** Questions or checks evaluated by the computer as True or False to decide what instruction to execute next (e.g., Is wake-up time equal to the current time?, Is Ground equal to SOIL?).
4. **Output:** Information sent out of the computer system to the user after processing (e.g., displaying text on a screen, an alarm ringing, a chatbot saying a sentence).

D. Evaluation & Success Criteria

- **Evaluation:** Evaluation means checking if the program is meeting the success criteria (expected outcome).
- **Success Criteria:** Success criteria clearly defines what is expected out of the program.

Section 2: Question and Answers

Q1: What does it mean to make an algorithm efficient?

Answer: An algorithm is efficient when it solves a problem in a careful, complete way using only the necessary instructions. This prevents any waste of time or computing steps.

Q2: How does selection make an algorithm more efficient than a linear sequence?

Answer: A linear algorithm carries out every single instruction every time, regardless of the situation. Selection uses conditions (IF... THEN... ELSE) to check whether an action is needed, skipping instructions that are unnecessary.

Q3: What is decomposition, and why is it useful in programming?

Answer: Decomposition is breaking down a large, complex problem into smaller, simpler sub-problems. It helps programmers plan clear solutions by organizing inputs, processes, conditions, and outputs step by step.

Q4: Define "Input" and "Output" in a computer system, with an example of each.

Answer: An input is data given to a computer by a user, example: typing a wake-up time on a keyboard.
An output is information or actions produced by the computer system after processing,
Example: such as an alarm ringing or text displayed on a screen.

Q5: What is a process in a program or algorithm?

Answer: A process is the set of steps carried out in order when a program or algorithm runs.
Example: An instruction or calculation, or assigning a typed answer to a variable.

Q6: What is success criteria?

Answer: Success criteria clearly defines what is expected out of the program.

Q7: What is a program evaluation?

Answer: Evaluation means checking if the program is meeting the success criteria (expected outcome).

Q8: What is a user prompt?

Answer: User prompt is when the program is waiting for the input from the user.

Q9) Deconstruct the following smart thermostat system into its exact Input, Process, Condition, and Output components:

- **User sets / enters the target temperature (22°C) via the dial.** []
- **The system stores the target temperature (22°C) into a variable.** []
- **Checking IF RoomTemperature < TargetTemperature (22°C).** []
- **Heater switches ON (or remains off if False).** []