Exercise

📘 A. Multiple Choice Questions (MCQs) – Answers with Explanations

1. **Which of the following best defines computational thinking?**
   ✅**(b)** A problem-solving approach that employs systematic, algorithmic, and logical thinking.

   🔹 *Explanation:* Computational thinking involves solving problems step-by-step, using logic, patterns, and algorithms.
2. **Why is problem decomposition important in computational thinking?**
   ✅**(a)** It simplifies problems by breaking them down into smaller, more manageable parts.

   🔹 *Explanation:* Decomposition makes a complex task easier to understand and solve.
3. **Pattern recognition involves:**
   ✅**(a)** Finding and using similarities within problems

   🔹 *Explanation:* It helps identify repeating elements to create efficient solutions.
4. **Which term refers to the process of ignoring the details to focus on the main idea?**
   ✅**(c)** Abstraction

   🔹 *Explanation:* Abstraction removes extra information and focuses on what is important.
5. **Which of the following is a principle of computational thinking?**
   ✅**(b)** Problem simplification

   🔹 *Explanation:* Simplifying problems helps in finding solutions more easily.
6. **Algorithms are:**
   ✅**(c)** Step-by-step instructions for solving a problem

   🔹 *Explanation:* An algorithm gives a clear path to solve a specific problem.
7. **Which of the following is the first step in problem-solving according to computational thinking?**
   ✅**(b)** Understanding the problem

   🔹 *Explanation:* You must understand the problem before you can plan a solution.
8. **Flowcharts are used to:**
   ✅**(b)** Represent algorithms graphically

   🔹 *Explanation:* Flowcharts visually show the steps in a process or algorithm.
9. **Pseudocode is:**
   ✅**(b)** A high-level description of an algorithm using plain language

   🔹 *Explanation:* Pseudocode is written in simple English to explain logic before actual coding.
10. **Dry running a flowchart involves:**
    ✅**(b)** Testing the flowchart with sample data

    🔹 *Explanation:* Dry running means checking the steps of an algorithm using test inputs to see if it works correctly.

📓 B. Short Questions – Simplified Answers

1. **Define computational thinking.**
   ➢ It is a logical and step-by-step method to solve problems using techniques like decomposition, pattern recognition, abstraction, and algorithms.
2. **What is decomposition in computational thinking?**
   ➢ It means breaking a big problem into smaller, easier parts to understand and solve.
3. **Explain pattern recognition with an example.**
   ➢ It is finding similarities in problems. For example, solving multiple math questions that use the same formula.
4. **Describe abstraction and its importance in problem-solving.**
   ➢ Abstraction means removing extra details and focusing on the main idea. It makes problem-solving easier.
5. **What is an algorithm?**
   ➢ An algorithm is a step-by-step list of instructions to solve a problem.
6. **How does problem understanding help in computational thinking?**
   ➢ Understanding the problem clearly helps in choosing the right method and solution.
7. **What are flowcharts and how are they used?**
   ➢ A flowchart is a diagram that shows the steps of an algorithm using symbols. It helps in understanding the process visually.
8. **Explain the purpose of pseudocode.**
   ➢ Pseudocode helps write an algorithm in plain English before converting it into actual programming code.
9. **How do you differentiate between flowcharts and pseudocode?**
   ➢ Flowcharts use shapes and arrows to show steps; pseudocode uses simple text. Both help in planning solutions.
10. **What is a dry run and why is it important?**
    ➢ Dry run is testing an algorithm using sample input to check if it works properly before coding.
11. **Describe LARP and its significance in learning algorithms.**
    ➢ LARP stands for Learn, Act, Reflect, Play. It helps students learn algorithms through practical activities, role-play, and reflection.
12. **List and explain two debugging techniques.**
    ➢ (1) **Dry Run:** Check steps manually.
    ➢ (2) **Print Statements:** Add outputs at different stages to find errors.

📓 C. Long Questions – Detailed Answers

1. Write an algorithm to assign a grade based on marks obtained:

Algorithm:

1. Start
2. Input marks
3. If marks ≥ 90 → Grade = A+
4. Else if marks ≥ 80 → Grade = A
5. Else if marks ≥ 70 → Grade = B
6. Else if marks ≥ 60 → Grade = C
7. Else → Grade = F
8. Display grade
9. End

---

2. Use algorithm design methods (flowcharts & pseudocode) to solve a problem

Example: Find if a number is even or odd

Pseudocode:

```
1. Start
2. Input number
3. If number mod 2 = 0
       → Print "Even"
     Else
       → Print "Odd"
4. End
```

Flowchart:

→ Start → Input Number → Check (Number % 2 == 0?)

Yes → Print "Even"

No → Print "Odd" → End

---

3. Define computational thinking and its significance.

Definition:

Computational thinking is a way of solving problems using logical steps, algorithms, and simplification techniques.

Significance:

- Used in science, business, and daily life.
- Helps create efficient and reusable solutions.
- Example: Using a recipe (algorithm) to cook a dish.

4. Explain decomposition and its importance.

Definition:
Decomposition means breaking a big task into smaller parts.

Importance:

- Makes complex problems easier to manage.
- Helps divide work in teams.
- Example: In making a website – design, code, test.

5. Explain pattern recognition in computational thinking.

Definition:
Pattern recognition is identifying similarities or trends in problems.

Importance:

- Helps solve problems faster.
- Useful in AI, math, and coding.

Example: Recognizing how passwords must include a capital letter or number.

6. What is abstraction? Why is it important?

Definition:
Abstraction is removing unneeded details to focus on the main task.

Importance:

- Reduces confusion
- Speeds up problem-solving
- Example: Ignoring background apps while focusing on typing a document.

7. What is an algorithm? Give an example and flowchart.

Definition:
An algorithm is a set of instructions to solve a task.

Example: Check if a number is positive or negative

Pseudocode:

```
Input number
If number > 0 → Print "Positive"
Else if number < 0 → Print "Negative"
Else → Print "Zero"
```

Flowchart:
Start → Input → Check number → Print result → End

---

8. Compare flowcharts and pseudocode.

Feature	Flowchart	Pseudocode
Format	Diagram with shapes	Text in plain English
Visual	Yes	No
Ease of writing	Slightly complex	Easy
Use	Planning, presentation	Planning, before coding
Best for	Beginners to visualize logic	Programmers to convert into code

---

9. Explain dry run and its role in validation.

Definition:
Dry run is checking an algorithm by using test inputs manually.

Importance:

- Helps detect logic errors
- Saves time before coding
- Works for both flowcharts and pseudocode

---

10. What is LARP? Its importance in algorithms.

LARP = Learn, Act, Reflect, Play

Importance:

- Makes learning interactive
- Helps students understand through role-play and teamwork
- Improves memory and logic

11. How does LARP support computational thinking?

Explanation:
LARP teaches problem-solving by encouraging students to:

- Act out algorithm steps
- Think logically in groups
- Reflect on mistakes

Example: Students play roles in a sorting algorithm to learn how data is arranged.