COMPUTER

7.1 Introduction to Computational Thinking

Q1: What is computational thinking?
Ans: Computational thinking is a method of solving problems using logical steps, breaking problems into smaller parts, and creating step-by-step solutions like a computer.

Q2: Why is computational thinking important?
Ans: It helps in solving complex problems easily, improves logical thinking, and is useful in programming and other fields.

7.2 Key Elements of Computational Thinking

Q3: What are the four main elements of computational thinking?
Ans: The four main elements are:

1. **Decomposition** – Breaking a big problem into smaller parts.
2. **Pattern Recognition** – Finding similarities or patterns in problems.
3. **Abstraction** – Focusing on important information only.
4. **Algorithms** – Creating step-by-step instructions to solve problems.

7.3 Decomposition

Q4: What is decomposition in computational thinking?
Ans: Decomposition means dividing a complex problem into smaller, easier parts to understand and solve them one by one.

Q5: Give an example of decomposition.
Ans: Making tea can be decomposed into steps like boiling water, adding tea leaves, adding milk, and serving.

7.4 Pattern Recognition

Q6: What is pattern recognition?
Ans: Pattern recognition means identifying repeated patterns or common solutions in problems to solve them faster.

Q7: Why is pattern recognition useful?
Ans: It helps to reuse solutions, saves time, and makes it easier to understand problems.

7.5 Abstraction

Q8: What is abstraction in computational thinking?
Ans: Abstraction is ignoring unnecessary details and focusing on the most important information needed to solve the problem.

Q9: Give an example of abstraction.
Ans: When drawing a map, only important things like roads and landmarks are shown, not every tree or building.

7.6 Algorithms

Q10: What is an algorithm?
Ans: An algorithm is a set of step-by-step instructions used to solve a problem or perform a task.

Q11: Give an example of an algorithm.
Ans: A recipe for baking a cake is an algorithm. It has steps like mixing ingredients, baking, and cooling.

7.7 Real-World Applications of Computational Thinking

Q12: How is computational thinking used in real life?
Ans: It is used in:

- **Daily tasks** like planning a trip.
- **Education** to solve math or science problems.
- **Programming** to create software.
- **Business** for data analysis and decision making.

7.8 Programming

Q13: How is computational thinking related to programming?
Ans: Computational thinking helps programmers break problems into smaller parts, create algorithms, and write code to solve problems effectively.

Q14: Why is computational thinking important for writing good programs?
Ans: It helps in writing clear, logical, and error-free code by planning the solution before starting the program.

7.9 Flowcharts

Q15: What is a flowchart?
Ans: A flowchart is a diagram that shows the steps of a process using symbols like rectangles, diamonds, and arrows.

Q16: Why are flowcharts useful?
Ans: Flowcharts help to:

- Understand the logic of a program.
- Identify errors easily.
- Communicate ideas clearly.

Q17: What are the common symbols used in flowcharts?
Ans:

- **Terminator (Oval):** Start or End
- **Process (Rectangle):** Any action or task
- **Decision (Diamond):** A yes/no question or condition
- **Arrow:** Direction of the flow

7.10 Pseudocode

Q18: What is pseudocode?
Ans: Pseudocode is a simple way of writing steps of an algorithm using plain language instead of programming syntax.

Q19: Why is pseudocode used?
Ans: It helps to plan a program's logic before writing the actual code. It is easy to read and understand.

Q20: What are some basic rules for writing pseudocode?
Ans:

- Write in simple English.
- Use keywords like START, IF, THEN, ELSE, END.
- Keep steps clear and logical.

7.11 Benefits of Computational Thinking

Q21: What are the benefits of computational thinking?
Ans:

- Improves problem-solving skills
- Helps in all subjects, not just computer science
- Encourages creativity and logical thinking
- Makes programming easier
- Useful in daily life and careers

7.12 Summary of the Chapter

Q22: What did you learn in this chapter?
Ans: In this chapter, we learned:

- What computational thinking is
- Its four main elements: decomposition, pattern recognition, abstraction, and algorithms
- How to use flowcharts and pseudocode to plan solutions
- How these skills are helpful in programming and real life

✓ Definitions (For Quick Revision)

1. **Computational Thinking:** A way of solving problems using logic, algorithms, and breaking tasks into smaller parts.
2. **Decomposition:** Breaking a problem into smaller parts.
3. **Pattern Recognition:** Finding similarities or repeated patterns.
4. **Abstraction:** Ignoring unimportant details and focusing on important information.
5. **Algorithm:** A step-by-step procedure to solve a problem.
6. **Flowchart:** A diagram that shows the steps of a process.
7. **Pseudocode:** A way of writing an algorithm in plain language.
