1. Exam Structure (Know This First)

Sections

1. **Part I – Direct Questions**
    - 5 short questions
    - **5 marks total** (1 mark each)
    - Usually: definitions, basic concepts, short explanations
2. **Part II – Calculations**
    - 5 questions
    - **15 marks total** (e.g. 3 marks each)
    - Focus:
        - Little’s Law
        - Amdahl’s Law
        - MFLOPS & time
        - Normalized performance & geometric mean
3. **Part III – Case Study**
    - 1 longer question
    - **10 marks**
    - You read a scenario and then explain/analyze performance, bottlenecks, or design choices.

2. Topics to Revise (Theory)

From the 3 chapters, make sure you cover at least:

2.1 CPU Basics & Execution

- CPU components: **ALU, Control Unit, Registers**
- **Instruction Cycle / Fetch–Decode–Execute**:
    1. Fetch instruction from memory
    2. Decode opcode & operands

3. Execute (ALU / memory / branch)
4. Write back result / update PC

- Types of instructions: data movement, arithmetic/logic, control (branch/jump).

2.2 Performance Concepts

- **Execution time** = how long a program runs
- **Throughput** = how many tasks per unit time
- **Latency** = time for one request / operation
- **CPI (Cycles Per Instruction)** and **Clock rate** (if you did this)
- Bottlenecks: slow memory, I/O, non-parallelizable code.

2.3 Performance Optimization & Laws

- Increasing clock speed, better architecture, parallelism
- **Little's Law** (queue/performance)
- **Amdahl's Law** (limits of speedup)
- **MFLOPS** (floating point performance)
- **Normalized performance** and **geometric mean** (comparing CPUs across multiple programs)