Placement Drive - Test Pattern for Software Developer Role

Stage-wise Test Pattern:

Round 1: Aptitude + C Programming MCQs

- **Type:** Objective (MCQ)
- **Duration:** 60 minutes
- **Number of Questions:** 20
- Topics Covered:
  - Quantitative Aptitude: Profit & Loss, Time-Speed-Distance, Permutations, Probability, etc.
  - Logical Reasoning: Puzzles, Series, Directions, Statements
  - Verbal Reasoning
  - Basic C Programming (pointers, loops, conditionals, output prediction)
  - Basic Java Programming
- **Difficulty:** Medium

Round 2: Basic Programming

- **Type:** Hands-on Coding
- **Duration:** 180 minutes
- **Number of Questions:** 5
- **Languages Allowed:** C / C++ / Java / Python
- Problem Types:
  - String manipulations (e.g., reverse, remove duplicates)
  - Pattern printing
  - Number theory problems (palindrome, prime, Armstrong)
  - Arrays and sorting logic
  - Control flow and loops
- **Expected Output:** Exact match with test cases

- **Difficulty:** Medium

Round 3: Advanced Programming

- **Type:** Hands-on Coding (Build-Logic Application)
- **Duration:** 60 minutes
- **Number of Questions:** 1–2
- **Languages Allowed:** C / C++ / Java
- Problem Types:
  - Mini projects (e.g., text editor, parser, menu-driven programs)
  - File handling, custom data structures
  - Object-Oriented Programming, modular design
  - Optimization-focused tasks
- **Evaluation:** Clean code, logic clarity, modularity, scalability
- **Difficulty:** High

Technical Interview Rounds:

Technical Interview 1:

- **Focus Areas:**
  - Data Structures (arrays, linked lists, stacks, queues)
  - OOP Concepts
  - Projects discussion
  - Code walk-through

Technical Interview 2 (if applicable):

- **Focus Areas:**
  - Code debugging and optimization
  - Low-Level Design (LLD)
  - System thinking and problem decomposition

HR Interview:

- **Topics:**
  - Career goals
  - Company fit and culture
  - Salary and job location
  - Availability and flexibility

Key Highlights:

- No negative marking
- Long test duration (~5 hours total)
- Focus on clean coding and logical clarity
- Equal weightage to aptitude and programming

Preparation Tips:

- Practice solving medium-level DSA problems
- Work on endurance coding (3–4 hour sessions)
- Master C/C++ basics, file handling, and OOP
- Mock interviews to improve articulation and debugging