

CSM TECH HACKATHON 2026

Logic • Design • Problem Solving Excellence
Technical Guidelines & Participation Rules

Hackathon Objective

This hackathon is designed to assess participants on **core programming fundamentals, logical reasoning, problem-solving ability, and clean technical design**. The focus is **not on frameworks or libraries**, but on **how effectively you think, structure, and implement solutions** under given constraints.

Evaluation Focus Areas

Participants will be evaluated on:

- Logical correctness and edge-case handling
- Adherence to given constraints
- Code clarity, readability, and structure
- Efficient use of fundamentals (loops, conditions, OOPS)
- Output accuracy and consistency

 **Short, clean, and logically sound solutions will always score higher than lengthy or over-engineered ones.**

Technical Guidelines

Programming Language

- participants must stick to their chosen language.
-

Constraint Compliance (Mandatory)

Each challenge will clearly specify constraints such as:

- Prohibited use of collections, strings, or libraries
- Mandatory use of arrays or arithmetic logic
- Object-Oriented design requirements

 **Violation of constraints will lead to partial or zero marks**, even if the output is correct.

Code Design Rules

- No hard-coded values unless explicitly mentioned
 - Avoid redundant logic and unnecessary variables
 - Follow proper naming conventions
 - Use modular and reusable logic wherever applicable
-

Object-Oriented Challenges

- Strict use of **Encapsulation, Inheritance, Polymorphism, and Abstraction**
 - Business logic must reside inside relevant classes
 - No salary, power, or calculation logic in the main method
-

Input & Output Handling

- Inputs must be handled exactly as specified in the problem statement
 - Output format must match the expected output **word-to-word**
 - Additional text or formatting will be treated as incorrect output
-

Edge Case Handling

- All boundary conditions mentioned in the problem must be handled
 - Missing edge cases will reduce evaluation score
-

Strictly Prohibited

- Use of AI tools, online compilers, or external references
 - Copy-paste from internet or peer solutions
 - Use of built-in shortcuts that bypass logic (e.g., reverse(), sort(), streams)
 - Discussion of solutions during the hackathon window
-

Time & Submission Rules

- Total Duration: **3 Hours**
 - Submit **only the source code** as instructed
 - Late submissions will not be evaluated
-

Scoring Philosophy

Category	Weightage
Correct Output	High
Logical Clarity	Very High
Constraint Compliance	Mandatory
Code Quality	High
Optimization	Bonus

💡 **Cleaner logic beats clever hacks. Readability beats shortcuts.**