



STEPCONE-2024

STANDARD OPERATING PROCEDURE

EVENT NAME: CODE WARS

DEPARTMENT NAME: CSE & CSE-AIML & AIDS

EVENT TYPE: Technical Event

EVENT DESCRIPTION:

The CODE WARS is a comprehensive coding competition that assesses various skills, including problem-solving, efficiency, and the ability to code under time pressure. This event will be hosted in ACM CODESPACE a competitive coding platform which is developed in our college. Participants progress through different levels based on their performance in each round, creating a competitive and engaging environment. The first round is “Code-A-Thon” where participants are given 8 coding questions and the score is evaluated based on some test cases passed. The second round “Passing Code” comprises 6 coding questions that must be solved before the question passes through the screen horizontally. This round adds a time constraint, testing participants' ability to code quickly and efficiently. The third round “Code Duel” is one code battle where each participant faces off with another participant and the one who solves is promoted to the next level of code battle.

EVENT GUIDELINES:

- All the team members registered should be present at the venue by the start time
- There should be no malpractice involved in working on the problem statement
- The participants should be present at the venue for the whole event (mentioned timings)
- Each participant should have at least one laptop

TEAM SIZE: 1 member

PLAN OF ACTION:

- Welcoming participants
- Introducing them to our coding interface.
- Our event has three rounds.
- We will Start the Round-1 followed by Round-2 and Round-3.
- The problem statements should be given based on the following key factors:-
 1. Clarity and Precision: The problem statement should be clear and precise, leaving no room for ambiguity in understanding the requirements.



2. **Input and Output Specification:** Clearly define the format of the input data and the expected format of the output. Specify any constraints on input values.
3. **Example Cases:** Provide illustrative examples to demonstrate how the solution should behave with different inputs. These examples help participants understand the problem better.
4. **Problem Constraints:** Clearly state any limitations or constraints on the solution, such as time complexity, space complexity, or specific algorithmic requirements.
5. **Scoring System:** If applicable, explain how the solution will be scored. This could include considerations for correctness, efficiency, and handling edge cases.

JUDGING CRITERIA:

The evaluation is done by the software based on various factors such as time constraints, test cases passed, etc.

