General Guidelines

1. **Week 6 will be a revision lab** covering all topics from **Week 1 to Week 5**.
2. **Pseudocode Requirement:**
    - Before coming to the lab, students must write the pseudocodes for the experiments listed below.
    - These pseudocodes will serve as a guide for implementation during the lab.
3. **No Internet Access:**
    - Ensure all required materials and references are prepared beforehand, as internet access will not be available in the lab.
4. **Weekly Assignments:**
    - Every week, the lab questions will be shared in advance.
    - Students must **write the pseudocode before coming to the lab** and **implement the code during the lab**.

Experiments (Pseudocode to be Prepared Beforehand)

1. **Implement the CartPole & Mountain Car Environment for a Certain Number of Steps**
    - Initialize the environment
    - Run for a predefined number of steps
    - Apply random actions or predefined policies
    - Record the results
2. **Implement the CartPole & Mountain Car Environment for a Certain Number of Episodes**
    - Initialize the environment
    - Loop through a set number of episodes
    - Reset environment at the start of each episode
    - Apply random actions or predefined policies
    - Store performance metrics
3. **Breadth-First Search (BFS)**
    - Initialize queue with the start node
    - Explore neighbors level by level
    - Keep track of visited nodes to avoid redundancy
    - Stop when the goal is found

4. **Depth-First Search (DFS)**
    - Initialize stack with the start node
    - Explore as deep as possible before backtracking
    - Keep track of visited nodes
    - Stop when the goal is found
5. **Theta Policy** (CartPole)
    - If the pole angle (θ) is **tilted left (θ < 0)**, push the cart left.
    - If the pole angle (θ) is **tilted right (θ > 0)**, push the cart right.
6. **Omega Policy** (CartPole)
    - If the pole's angular velocity (ω) is **moving away from vertical (ω < 0)**, push left.
    - If the pole's angular velocity (ω) is **moving towards vertical (ω > 0)**, push right.
7. **Simple Hill Climbing**
    - Start with a random solution
    - Evaluate its fitness
    - If a better neighboring solution exists, move to it
    - Repeat until no better neighbors are found
8. **Stochastic Hill Climbing**
    - Start with a random solution
    - Select a random neighbor instead of the best one
    - Move to the neighbor if it improves the fitness
    - Repeat the process
9. **Steepest Ascent Hill Climbing**
    - Start with a random solution
    - Evaluate all possible neighbors
    - Move to the best possible neighbor
    - Repeat until no better neighbors are found

Note: Ensure you bring the pseudocode for all the above experiments before attending the lab. Implementation will be done in the lab session itself.