Department of CS/CSE/CSIT

AI & It's Application (CS205B)

MSE-2-Instructions for the students

1. Pre-Hackathon Phase (To be completed by 14th Nov-2025)

Activities:

- **Announcement (By the Concerned Dept/ Faculty)**
  - Share the Tracks/Themes, date, and evaluation criteria. Out of these themes, student can select either one theme, specific project would be assigned on the day of Hackathon
  - Form teams (5 students each).
  - Submit your team details to the AI teacher by 14th Nov-2025.
- **Orientation Session**
  - Rules and evaluation parameters
  - Though it's a group activity still marks would be awarded based on individual performance in the hackathon.

---

2. Hackathon Day :

A. Branch and Section wise Timings: 21st Nov-2025

Shift-1(9:30 AM-12:30 PM)	Shift-2(1:30 PM-4:30PM)
CS-A, CS-B	CS-C, CS-D
CSE-A, CSE-B	CSE-C, CSE-D, CSE-E
CSIT-A, CSIT-B	CSIT-C

Actual Seating plan would be shared by Exam cell.

Duration: 3 Hours

B. Hackathon Stages

Stage	Title	Duration	Output
Stage 1	Problem Definition & Environment Selection	30 mins	Problem statement & chosen track
Stage 2	Agent Architecture Design	30 mins	Agent interaction diagram & logic plan
Stage 3	Implementation	1 hour	Working prototype (simulation-ready)
Stage 4	Testing, Optimization & Collaboration Tuning	15 mins	Final simulation run & performance logs

Stage	Title	Duration	Output
Stage 5	Presentation Prep, Evaluation & Feedback	45 mins	Live demo + presentation

C. Evaluation Criteria (Total 40 marks)

Category	Marks	Description
Problem Understanding	8	Clarity of approach & AI concept relevance
Agent Performance	8	Accuracy, precision, or other chosen metrics
Innovation	8	Creativity in applying AI techniques
Presentation & Justification	8	Clarity, visualization, explainability
Team Collaboration & Code Quality	8	Balanced effort, modular & documented code

---

Hackathon Themes & Topics

Track 1: Task-Cooperative Agents

Theme: Agents that help each other achieve shared goals (focus on cooperation, planning, search, and logic).

1. Dual Maze Navigators

Objective: Implement two cooperative agents that navigate a maze to collect all keys.

- **Description:** The maze has scattered keys. Each agent explores different regions using BFS/DFS to collect them faster.
- **Hint:** Use a shared grid representation and track visited cells; coordinate to avoid overlap.
- **Output:** Final maze visualization with both agents' paths.
- **Bonus:** Add communication for path updates.

2. Cleaning Crew Coordination

Objective: Design two cleaning bots that divide rooms and clean efficiently.

- **Description:** The environment is a 2D grid with dirty cells. Agents plan non-overlapping paths.
- **Hint:** Implement A* or Greedy Search for path planning; coordinate via shared task lists.
- **Output:** Visualization showing cleaned cells and efficiency score.

3. Cooperative Path Planners

Objective: Build two agents that must reach two goals while avoiding collisions.

- **Hint:** Each agent uses A* with a shared collision-avoidance logic.
- **Output:** Grid animation showing synchronized movement.

4. Warehouse Pickup Team

Objective: Agents must pick and drop items in a warehouse grid cooperatively.

- **Hint:** Assign pick-up tasks based on proximity; minimize total travel distance.
- **Output:** Table showing total time and efficiency metric.

5. Rescue Bot Squad

Objective: Agents find and rescue trapped victims in a maze.

- **Hint:** BFS for exploration, logic-based assignment for rescue zones.
- **Output:** Visualization of agents rescuing victims.

6. Dual Drone Delivery

Objective: Two drones deliver packages to different locations with minimal overlap.

- **Hint:** Use A* search and assign packages using a greedy or Hungarian algorithm.
- **Output:** Total delivery time and heatmap of coverage.

7. Grid Painting Agents

Objective: Two robots paint cells without overlapping.

- **Hint:** Use DFS or rule-based task allocation.
- **Output:** Color-coded grid showing paint coverage.

8. Resource Collection Team

Objective: Collect resources scattered in a map cooperatively.

- **Hint:** Shared task queue and distributed decision logic.
- **Output:** Plot showing resources collected by each agent.

9. Cooperative Firefighters

Objective: Agents must extinguish fires in different zones efficiently.

- **Hint:** Use BFS for fire spread simulation, cooperative task allocation.
- **Output:** Graph showing total time to extinguish fires.

10. Map Exploration Partners

Objective: Agents explore unknown regions together.

- **Hint:** Divide unexplored regions using grid partitioning logic.
- **Output:** Heatmap showing exploration efficiency.

Track 2: Competitive Agents

Theme: Agents compete to win — introducing simple **game theory**, **reinforcement**, or **strategic learning**.

1. Treasure Grab

Objective: Two agents race to collect the most coins in a maze.

- **Hint:** Use BFS for movement; score = coins collected.
- **Output:** Final score table, grid animation.

2. Capture the Flag

Objective: Competing agents defend their flag while trying to steal the opponent's.

- **Hint:** Combine A* pathfinding + defensive zone logic.
- **Output:** Final winner and captured flag logs.

3. Resource War

Objective: Competing agents gather limited resources; once taken, unavailable.

- **Hint:** Shared environment; random resource spawn.
- **Output:** Resource collection statistics.

4. Smart Snake AI Duel

Objective: Competing snake agents play to survive and block the other.

- **Hint:** Use rule-based logic or simple Q-learning.
- **Output:** Game replay and scores.

5. Grid Racing AI

Objective: Competing agents race from start to goal through obstacles.

- **Hint:** BFS/A* with random obstacle generation.
- **Output:** Completion time comparison.

6. Market Trader Duel

Objective: Two agents buy/sell commodities for profit.

- **Hint:** Rule-based decision logic or reinforcement learning.
- **Output:** Profit chart for both agents.

7. Predator-Prey Simulation

Objective: One agent (predator) hunts; other (prey) escapes.

- **Hint:** Greedy distance-based movement.
- **Output:** Simulation of chase outcome.

8. Maze Domination

Objective: Agents occupy the most territory in a limited time.

- **Hint:** Grid expansion game with conflict resolution logic.
- **Output:** Final occupied area percentage.

9. Competitive Cleaner

Objective: Agents clean rooms but compete for higher score.

- **Hint:** Shared environment, scoring by area cleaned.
- **Output:** Leaderboard visualization.

10. Bidding War AI

Objective: Competing agents bid for limited items.

- **Hint:** Use random or rule-based bidding strategy.
- **Output:** Bidding history and winner summary.

Track 3: Smart City Agents

Theme: Agents manage and optimize aspects of smart city life collaboratively.

1. Traffic Light Optimizer

Objective: Build traffic light agents to reduce average waiting time.

- **Hint:** Use rule-based logic or reinforcement updates.
- **Output:** Average delay graph before vs after optimization.

2. Garbage Collection Routing

Objective: Agents plan routes to collect waste efficiently.

- **Hint:** Use BFS or Dijkstra for route planning.
- **Output:** Total travel distance and time comparison.

3. Smart Parking Allocation

Objective: Agents allocate parking dynamically.

- **Hint:** Use matching algorithm (Hungarian) for assignment.
- **Output:** Parking usage heatmap.

4. Energy Distribution Agents

Objective: Balance energy supply among buildings.

- **Hint:** Implement rule-based balancing logic.
- **Output:** Plot showing load balance efficiency.

5. Water Supply Optimizer

Objective: Agents manage water distribution to multiple houses.

- **Hint:** Constraint satisfaction for pressure and flow balance.
- **Output:** Water flow distribution chart.

6. Emergency Response Dispatchers

Objective: Agents assign ambulances/fire trucks to incidents.

- **Hint:** Use nearest-neighbor or greedy optimization.
- **Output:** Average response time.

7. Pollution Control Monitors

Objective: Agents monitor pollution and coordinate mitigation.

- **Hint:** Grid simulation + distributed alert logic.
- **Output:** Graph of pollution levels over time.

8. Streetlight Energy Saver

Objective: Streetlight agents dim/brighten adaptively.

- **Hint:** If sensors detect movement, increase brightness.

- **Output:** Total energy saved metric.

9. Smart Bus Routing

Objective: Bus agents plan optimal city routes.

- **Hint:** Graph representation + Dijkstra-based optimization.
- **Output:** Total passenger wait time reduction.

10. Waste Segregation AI

Objective: Agents classify and sort waste items.

- **Hint:** Use simple ML classifier with scikit-learn.
- **Output:** Classification accuracy.

Track 4: Communication & Negotiation Agents

Theme: Agents communicate, plan, or negotiate for common/limited goals.

1. Message Passing Maze

Objective: Agents exchange messages to locate a hidden treasure.

- **Hint:** Use BFS with communication rules.
- **Output:** Path visualization showing message flow.

2. Resource Negotiators

Objective: Agents negotiate resource sharing.

- **Hint:** Use simple rule-based bidding/negotiation.
- **Output:** Negotiation logs.

3. Task Division through Communication

Objective: Agents divide multiple tasks using short messages.

- **Hint:** Use message-passing to allocate subtasks.
- **Output:** Efficiency of task completion.

4. Chat-to-Plan Navigators

Objective: Agents plan joint paths through text-based negotiation.

- **Hint:** Each agent proposes next step; others vote/approve.
- **Output:** Conversation + final path visualization.

5. Multi-Agent Auction

Objective: Agents bid via messages for task ownership.

- **Hint:** Implement simple auction protocol.
- **Output:** Winning bids and profit.

6. Disaster Relief Coordinators

Objective: Agents communicate to distribute resources effectively.

- **Hint:** Message-based coordination in grid world.
- **Output:** Graph showing resource delivery coverage.

7. Messenger Chain AI

Objective: Pass a message across multiple agents with minimal hops.

- **Hint:** Implement shortest communication chain.
- **Output:** Message passing efficiency.

8. Negotiating Cleaners

Objective: Cleaners discuss zone allocation to avoid overlap.

- **Hint:** Negotiation-based decision logic.
- **Output:** Conflict-free task completion log.

9. Language Evolving Agents

Objective: Agents develop a shared simple communication protocol.

- **Hint:** Random symbols for actions; evolve agreement.
- **Output:** Visualization of evolved vocabulary.

10. Delivery Talkers

Objective: Agents coordinate deliveries using short message exchange.

- **Hint:** Use encoded task messages and rule-based understanding.
- **Output:** Reduced delivery time.

For any clarification please connect with Your AI teacher