Building AI Agents – Artificial and Computational Intelligence

Course Description:

This 40-hour intensive program offers a hands-on, industry-relevant journey into the design and development of AI Agents using principles from Artificial and Computational Intelligence. Structured across 8 progressive modules, the course emphasizes agent-based systems, intelligent search, optimization strategies, knowledge representation, and probabilistic reasoning. Each module includes Python-based coding labs to translate theory into real-world applications, preparing learners to build AI agents capable of navigating complex environments and making intelligent decisions.

Audience Profile:

This course is ideal for:

- Aspiring AI developers and ML engineers
- Computer science and engineering students
- Software developers exploring intelligent systems
- Researchers, data enthusiasts, and tech professionals interested in agent-based AI

Prerequisites:

- Basic Python programming skills
- Understanding of algorithms and data structures
- Familiarity with mathematical concepts such as logic and probability

Course Objectives:

By the end of this course, participants will be able to:

- Understand the architecture and behavior of intelligent agents

- Implement uninformed and informed search strategies for pathfinding and problem-solving
- Apply local search and evolutionary algorithms for optimization
- Design AI systems for adversarial environments using game tree search
- Build rule-based reasoning systems using propositional logic
- Develop probabilistic models such as Bayesian Networks and Hidden Markov Models
- Translate theoretical AI concepts into functional, Python-based agents

---

Table of Contents (ToC):

Module 1: Introduction to AI and Intelligent Agents

- What is AI? Acting vs Thinking rationally/humanly
- AI application domains
- Concept of agents and environments
- Rationality and agent architectures
- *Hands-on*: Build a simple reflex-based agent (Grid World)

Module 2: Classical Search Techniques – Part 1

- Problem formulation
- Uninformed search: BFS, DFS, UCS
- *Hands-on*: Solve a maze using BFS or DFS in Python

Module 3: Classical Search Techniques – Part 2

- Informed search: Greedy, A*
- Heuristic design: Admissibility & consistency
- *Hands-on*: Pathfinding using A* with custom heuristics

Module 4: Optimization via Local Search

- Hill Climbing and Simulated Annealing
- Introduction to Genetic Algorithms (GA)

- Key parameters: selection, crossover, mutation
- *Hands-on*: GA for function optimization (e.g., Max Ones problem)

Module 5: Game Playing & Adversarial Search

- Minimax algorithm
- Alpha-Beta Pruning
- Introduction to real-time/imperfect decision games
- *Hands-on*: Build a Tic-Tac-Toe AI using Minimax

Module 6: Logical Reasoning and Knowledge Representation

- Propositional logic: syntax, semantics
- First-order logic (overview only)
- Forward & backward chaining
- Resolution method (brief)
- *Hands-on*: Rule-based logic system using PyKE or custom inference engine

Module 7: Probabilistic Reasoning & Bayesian Networks

- Need for probabilistic models
- Basics of Bayesian Networks: nodes, edges, CPTs
- Exact Inference: Enumeration
- Approximate Inference: Sampling (brief)
- *Hands-on*: Use pgmpy or pomegranate to build a basic BN

Module 8: Hidden Markov Models (HMMs)

- Introduction to temporal models
- HMM structure: transition, emission, observation
- Filtering, smoothing, Viterbi algorithm
- *Hands-on*: Predict weather/speech patterns using HMM in Python