Semester-VI

DEPARTMENT OF COMPUTER SCIENCE

[UG Programme for Bachelor in Computer Science (Honours)]

DISCIPLINE SPECIFIC CORE COURSE – 16 (DSC-16) : Artificial Intelligence

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC-16 Artificial Intelligence	**4**	**3**	**0**	**1**	Pass in Class XII	DSC 04 Object Oriented Programming with C++/ GE 1a Programming using C++ / GE1b Programming with Python/ DSC 01 Programming using Python/ GE 3b: Java Programming

Learning Objectives

The course objectives of this course are to:

- To introduce basic concepts and techniques of Artificial Intelligence (AI).
- To apply informed search techniques for different applications.
- To learn various knowledge representation techniques and writing Prolog programs.
- To learn about the latest techniques for developing AI systems.

Learning outcomes

On successful completion of this course, students will be able to:

- identify problems that are amenable to solutions by specific AI methods.
- state the utility of different types of AI agents.
- apply different informed search techniques for solving real world problems.
- use knowledge representation techniques for AI systems.

SYLLABUS OF DSC-16

Unit 1 6 Hours

Introduction: Introduction to artificial intelligence, background and applications, Turing test, Weak AI, Strong AI, Narrow AI, Artificial General Intelligence, Super AI, rational agent

approaches to AI, introduction to intelligent agents, their structure, behavior and task environment.

Unit 2 12 Hours

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth-first search, depth-first search, hill climbing and its variations, heuristics search techniques: best-first search, A* algorithm, constraint satisfaction problem, means-end analysis, introduction to game playing, min-max and alpha-beta pruning algorithms.

Unit 3 16 Hours

Knowledge Representation: Propositional logic, First-Order Predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, and scripts, production rules, Introduction to Programming in Logic (PROLOG).

Unit 4 8 Hours

Understanding Natural Languages: Components and steps of communication, the contrast between formal and natural languages in the context of grammar, Chomsky hierarchy of grammars, parsing, and semantics, Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented transition nets.

Unit 5 3 Hours

AI The Present and the Future: Symbolic AI, Data-driven AI and Machine Learning, Introduction to Machine Learning and Deep Learning based AI, some applications of symbolic and data driven AI, Interpretable and Explainable AI, Ethics of AI: benefits and risks of AI.

Essential/recommended readings

1. Russell, Stuart, J. and Norvig, Peter, *Artificial Intelligence - A Modern Approach*, Pearson, 4th edition, 2020..
2. Bratko, Ivan, *Prolog Programming for Artificial Intelligence*, Addison-Wesley, Pearson Education, 4th edition, 2012.
3. Patterson, DAN,W, *Introduction to A.I. and Expert Systems* – PHI, 2007.
4. Clocksin, W., F. and Mellish, *Programming in PROLOG*, 5th edition, Springer, 2003.

Additional references

1. Kaushik, Saroj, *Artificial Intelligence*, Cengage Learning India, 2011.
2. Rich, Elaine and Knight, Kelvin, *Artificial Intelligence*, 3rd edition, Tata McGraw Hill, 2010

Practical List :

Practical exercises such as

1. Write a program in Prolog to implement TowerOfHanoi(N) where N represents the number of disks.
2. Write a program to implement the Hill climbing search algorithm in Prolog.
3. Write a program to implement the Best first search algorithm in Prolog.
4. Write a program to implement A* search algorithm in Prolog.
5. Write a program to implement the min-max search algorithm in Prolog.

6. Write a program to solve the Water-Jug Problem in Prolog.
7. Implement sudoku problem (minimum 9×9 size) using constraint satisfaction in Prolog.
8. Write a Prolog program to implement the family tree and demonstrate the family relationship.
9. Write a Prolog program to implement knowledge representation using frames with appropriate examples.
10. Write a Prolog program to implement conc(L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
11. Write a Prolog program to implement reverse(L, R) where List L is original and List R is reversed list.
12. Write a Prolog program to generate a parse tree of a given sentence in English language assuming the grammar required for parsing.
13. Write a Prolog program to recognize context free grammar $a^n b^n$.