## DSE-03 (b): 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 (if any) |
|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical/ Practice | | |
| Artificial Intelligence | 4 | 3 | 0 | 1 | Class XII | DSC-03 DSC-06 |

***Learning Objectives:***

*1. Study the concepts of Artificial Intelligence.*
*2. Learn the methods of solving problems using Artificial Intelligence.*
*3. Learn the knowledge representation techniques, reasoning techniques and planning*
*4. Introduce the concepts of Expert Systems and machine learning.*

***Learning Outcomes:***

*1. Identify problems that are amenable to solutions by specific AI methods.*
*2. Appreciate the utility of different types of AI agents.*
*3. Apply different informed search techniques for solving real world problems.*
*4. Use knowledge representation techniques for AI systems..*
*5. Understand human level, data driven and end to end approaches to AI.*

**UNIT-I** **(10 Hours)**

**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 , the Present and the Future of AI.

**UNIT-II** **(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-III** **(12 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-IV** **(11 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.

***References:***

*1. Stuart J. Russell and Peter Norvig, Artificial Intelligence - A Modern Approach, Pearson, 4th edition, 2020.*
*2. Elaine Rich and Kelvin Knight, Artificial Intelligence, 3 rd edition, Tata McGraw Hill, 2010.*
*3. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 4th edition, 2012.*

**List of Practicals:** **(30 hours)**

1. Write a prolog program to calculate the sum of two numbers.
2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.
3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N. 60
4. Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.
5. Write a Prolog program to implement GCD of two numbers.
6. Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.
7. Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.
8. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.
9. 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.
10. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.
11. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.
12. Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.
15. Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list.
16. Write a prolog program to implement insert_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.
17. Write a Prolog program to implement delete_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.
18. Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.