

MANIPAL UNIVERSITY JAIPUR

Department AIML

Course Hand-out

A. Basic Details:

Programme Name:	B.Tech CSE (AIML)
Course Name:	Automata Theory & Compiler Design
Course Code:	AIM3202
LTPC (<i>Lecture Tutorial Practical Credits</i>):	3 1 0 4
Session:	July–Dec 2025
Class:	V Semester
Course Coordinator:	Dr. Jaydeep Kishore
Course Instructor(s):	Dr. Upendra Singh, Dr. Shail, Dr. Jaydeep Kishore
Additional Practitioner(s) – if any (<i>Industry Fellow/ Visiting Faculty/ Adjunct Faculty, etc.</i>):	NA

B. Introduction:

This course introduces the foundational principles of Automata Theory and Compiler Design. It covers finite automata, context-free grammars, pushdown automata, Turing machines, and fundamental concepts in compiler design such as lexical analysis, parsing, syntax-directed translation, and code optimization. Students will gain theoretical and practical knowledge essential for designing language processors and understanding computation limits.

C. Course Outcomes:

<i>CO Statement</i>	<i>CO</i>	<i>Bloom's Level</i>	<i>Target Attainment %</i>	<i>Target Attainment level</i>
Understand and design finite automata, regular expressions, and their equivalence.	CO1	Understand (L2)	3	80%
Analyze and construct context-free grammars and pushdown automata.	CO2	Analyze (L4)	3	80%
Explain and simulate Turing Machines and evaluate decidability.	CO3	Evaluate (L5)	3	80%
Apply parsing techniques and syntax analysis in compiler design.	CO4	Apply (L3)	3	80%
Develop intermediate code, perform code optimization and generation.	CO5	Create (L6)	3	80%

C. Program Outcomes and Program Specific Outcomes:

Program Specific Outcomes (PSOs)

PSO1: Graduates will be able to examine the applications of Artificial Intelligence and Machine Learning in real-life problems.

PSO2: Graduates will be able to design and implement intelligent systems for multidisciplinary problems.

Program outcomes (POs)

D.

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and

write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Assessment Plan: Clearly write the criteria, its description, and associated marks for assessment of student achievements. In addition, the attendance requirements, assignments need to be mentioned in this section.

Criteria	Description	Maximum Marks
Internal Assessment (Summative)	Mid-Term Examination (Close Book)	30
	Class Work Sessional (CWS): Quiz (10), Assignment(10), Attendance (10)	30
End Term Exam (Summative)	End Term Exam (CloseBook)	40
	Total	100

E. Syllabus:

Introduction to Automata Theory: Alphabets, Strings, and Languages, Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), Equivalence of DFA and NFA, Regular Expressions and their equivalence with Finite Automata, Applications of Finite Automata.

Context-Free Grammars and Pushdown Automata: Context-Free Grammars (CFGs), Derivations and Parse Trees, Ambiguity in Grammars, Simplification of CFGs, Pushdown Automata (PDA) – Definition and Language Recognition, Equivalence of PDA and CFG.

Turing Machines and Computability: Introduction to Turing Machines, Language Acceptance by Turing Machines, Variants of Turing Machines, Church-Turing Thesis, Recursive and Recursively Enumerable Languages, Undecidability and the Halting Problem.

Introduction to Compiler Design: Structure of a Compiler, Lexical Analysis – Role, Specification of Tokens, Lex Tools, Syntax Analysis – Context-Free Grammars, Top-Down Parsing (Recursive Descent, LL(1)), Bottom-Up Parsing (Shift-Reduce, LR, SLR, LALR), Parse Trees and Abstract Syntax Trees (AST).

Semantic Analysis and Code Optimization: Intermediate Code Generation – Three Address Code, Syntax-Directed Translation, Type Checking and Symbol Table, Code Optimization, DAGs, Code Generation.

References:

1. Hopcroft, Ullman, and Motwani – *Introduction to Automata Theory, Languages and Computation*

2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman – *Compilers: Principles, Techniques, and Tools* (Dragon Book)
3. Peter Linz – *An Introduction to Formal Languages and Automata*
4. John C. Martin-*Introduction to Languages and the Theory of Computation.*
5. Dick Grune et al. – *Modern Compiler Design*

F. Lecture Plan:

Lecture No.	Topic	Session Outcome	Corresponding CO	Mode of delivery	Mode of assessing CO
1	Alphabets, Strings, Languages	Understand Alphabets,	CO1	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
2	DFA: definition and construction	Understand and design DFA:	CO1	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
3	Some Example of DFA	Understand and design DFA:	CO1	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
4	Some Example of DFA	Understand and design DFA:	CO1	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
5	Non-Deterministic Finite Automata (NFA)	Understand and design NFA:	CO1	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
6	Equivalence of DFA and NFA	Understand and design equivalence	CO1	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
7	Example NFA to DFA	Understand and design their conversion	CO1	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
8	Regular Expressions and their equivalence with Finite Automata	Understand and design RE	CO1	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
9	Example RE	Understand and design RE	CO1	Learning through	In Class Quiz Home

				problem-solving	Assignment Mid Term End Term
10	Arden's theorem	Understand and design the conversion from FA to RE	CO1	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
11	Applications of Finite Automata	Understand The applications	CO1	Peer Teaching	In Class Quiz Home Assignment Mid Term End Term
12	Grammar and Language	Analyze and construct grammars	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
13	Some example of grammar construction	Analyze and construct grammars	CO2	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
14	Some example of grammar construction	Analyze and construct grammars	CO2	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
15	Chomsky Classification of language	Analyze and construct grammars	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
16	Context-Free Grammars (CFGs)	Analyze and construct CFG	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
17	Derivations and Parse Trees	Analyze and construct Parse Tree	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
18	Ambiguity in Grammars	Analyze The ambiguity in grammars	CO2	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term
19	Simplification of CFGs	Analyze and construct the simplified CFG	CO2	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term

20	CNF	Analyze and construct the simplified CFG	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
21	GNF	Analyze and construct the simplified CFG	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
22	GNF Example	Analyze and construct the simplified CFG	CO2	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term
23	Pushdown Automata (PDA) – Definition and Language Recognition	Analyze and construct the PDA	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
24	Some Example of PDA	Analyze and construct the PDA	CO2	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term
25	Equivalence of PDA and CFG	Analyze and construct the relationship between CFG and PDA	CO2	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
26	Example PDA to CFG	Analyze and construct the relationship between CFG and PDA	CO2	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term
27	Introduction to Turing Machines	Explain and simulate Turing Machines	CO3	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
28	Language Acceptance by Turing Machines	Explain and simulate Turing Machines	CO3	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
29	Some Example of TM	Explain and simulate Turing Machines	CO3	Group-teaching and learning	In Class Quiz Home Assignment Mid Term End Term

30	Some Example of TM	Explain and simulate Turing Machines	CO3	Learning through problem-solving	In Class Quiz Home Assignment Mid Term End Term
31	Church-Turing Thesis, Recursive and Recursively Enumerable Languages	Explain and simulate Turing Machines	CO3	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
32	Undecidability and the Halting Problem	evaluate decidability.	CO3	Lecture based teaching-learning	In Class Quiz Home Assignment Mid Term End Term
Mid Term Examination					
33	Structure of a Compiler	Apply Automata Concept	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
34	Lexical Analysis Role, Specification of Tokens, Lex Tools	Apply FA in Lexical Analysis	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
35	Syntax Analysis Context-Free Grammars	Apply parsing techniques and syntax analysis in compiler design	CO4	Learning through problem-solving	In Class Quiz Home Assignment End Term
36	Top-Down Parsing (Recursive Descent, LL (1))	Apply parsing techniques and syntax analysis in compiler design	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
37	Top-Down Parsing (Recursive Descent, LL (1))	Apply parsing techniques and syntax analysis in compiler design	CO4	Learning through problem-solving	In Class Quiz Home Assignment End Term
38	Bottom-Up Parsing	Apply parsing techniques and syntax analysis in compiler design	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
39	Bottom-Up Parsing (Shift-Reduce, LR,	Apply parsing techniques and syntax	CO4	Lecture based	In Class Quiz Home Assignment

		analysis in compiler design		teaching-learning	End Term
40	SLR, LALR	Apply parsing techniques and syntax analysis in compiler design	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
41	Example SLR	Apply parsing techniques and syntax analysis in compiler design	CO4	Learning through problem-solving	In Class Quiz Home Assignment End Term
42	Example Bottom-Up Parsing	Apply parsing techniques and syntax analysis in compiler design	CO4	Learning through problem-solving	In Class Quiz Home Assignment End Term
43	Example Bottom-Up Parsing	Apply parsing techniques and syntax analysis in compiler design	CO4	Learning through problem-solving	In Class Quiz Home Assignment End Term
44	Parse Trees and Abstract Syntax Trees (AST).	Apply parsing techniques and syntax analysis in compiler design	CO4	Lecture based teaching-learning	In Class Quiz Home Assignment End Term
45	Three Address Code, Syntax-Directed Translation	Develop TAC	CO5	Flipped Classroom	In Class Quiz Home Assignment End Term
46	Type Checking and Symbol Table*	Develop type checking	CO5	Flipped Classroom	In Class Quiz Home Assignment End Term
47	Code Optimization*	perform code optimization and generation.	CO5	Peer teaching	In Class Quiz Home Assignment End Term
48	DAGs, Code Generation*	perform code optimization and generation.	CO5	Flipped Classroom	In Class Quiz Home Assignment End Term
End Term Examination					

G. Course Articulation Matrix:

CO	PO 1	PO 2	PO3	PO4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO 12	PSO 1	PSO 2
CO1	3	3	2	0	2	0	0	0	1	1	0	2	3	2
CO2	3	3	2	1	2	0	0	0	1	1	0	2	3	2
CO3	3	3	2	2	2	0	0	1	1	1	0	3	3	2
CO4	3	2	2	2	3	0	0	1	2	2	1	2	3	3
CO5	3	2	3	2	3	1	1	1	2	2	2	3	3	3

Course Co-Ordinator**Head of Dept.
CSE(AIML)****Student Representative
Name:
Registration No.:**