Course Title	**Theory of Automata**		
Course Code	**CSC-312**		
Credit Hours	3 (3,0)		
Category	Computer Science Core		
Prerequisite	None		
Co-Requisite	None		
Follow Up	CSE-326 Compiler Construction		
Course Introduction	This course introduces the fundamental concepts of automata theory and formal languages. It covers finite automata, regular expressions, grammars, pushdown automata, and Turing machines, focusing on their design, analysis, equivalence, and computational limitations. Applications in compiler design and language processing are also explored.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	**BT**	**PLO**
	CLO1: Explain the concepts of finite automata, regular expressions, and formal languages.	C2 (Understand)	1,2
	CLO2: Design automata to recognize specific languages.	C3 (Apply)	2,4
	CLO3: Demonstrate the equivalence of automata, regular expressions, and grammars.	C4 (Analyze)	2,3
	CLO4: Analyze the limitations of finite automata and context-free grammars.	C4 (Analyze)	2,3
	CLO5: Apply automata theory concepts in basic compiler components such as lexical analysis and pattern matching.	C3 (Apply)	2,4,5
Text Book(s)	Peter Linz, "An Introduction to Formal Languages and Automata", 6th Ed., Jones & Bartlett Publishers, 2016. ISBN: 1284077241, 978-1284077247		
Reference Material	Daniel I. A. Cohen, "Introduction to computer theory", 2nd Ed., Wiley India Pvt. Limited, 2007. ISBN 0471137723, 9780471137726		

Weekly Lesson Plan

Week No.	Course Content	Assignments/Readings
Week 1	Course introduction and background. Review of sets, formal languages, grammars, and automata. Introduction to the Chomsky hierarchy. Basics of Kleene closure and its properties.	Linz: Chapter 1
Week 2	Introduction to finite automata. Types of finite automata: acceptors and transducers. Deterministic finite automata (DFA) and their representation using transition graphs, tables, and functions. Introduction to regular languages.	Linz: Chapter 2
Week 3	Construction and analysis of deterministic finite automata (DFA). DFA constructions for union, concatenation, complement, and reversal of regular languages. Introduction to cross product construction for DFA.	Linz: Chapter 2 Quiz1
Week 4	Introduction and motivation for nondeterministic finite automata (NFA). Representation of NFA using transition graphs, tables, and transition functions. λ-NFA (epsilon-NFA) and its conversion to NFA.	Linz: Chapter 2 Quiz2

Week 5	Conversion of NFA to DFA. Equivalence of DFA and NFA. Minimization of deterministic finite automata (DFA).	Linz: Chapter 2
Week 6	Introduction to regular expressions and their formal definition. Operations on regular expressions: union, concatenation, and Kleene star.	Linz: Chapter 3 Quiz3
Week 7	Conversion of regular expressions to finite automata. Introduction to conversion of finite automata to regular expressions (state elimination method). Equivalence of regular expressions and finite automata. Applications of regular expressions.	Linz: Chapter 3 Quiz4
Week 8	Introduction to grammars. Regular grammars: production rules and their classification into right-linear and left-linear grammars. Equivalence of regular grammars with finite automata and regular expressions.	Linz: Chapter 3
Week 9	Finite automata with output (transducers): Moore and Mealy machines and their representation using transition graphs, tables, and transition functions. Equivalence of Moore and Mealy machines.	Linz: Chapter 2
Week 10	Pumping lemma for regular languages and its applications.	Linz: Chapter 4 Quiz5
Week 11	Introduction to pushdown automata (PDA). Motivation for PDA based on limitations of finite automata. Types of pushdown automata (PDA). Formal definition of PDA. Representation of PDA using transition diagrams and transition functions.	Linz: Chapter 7
Week 12	Introduction to context-free grammars (CFG). Components of CFG. Derivations and parse trees. Ambiguity in CFG.	Linz: Chapter 5 Quiz6
Week 13	Simplification of CFG (removal of useless symbols, null productions, and unit productions). Chomsky Normal Form (CNF) and Backus–Naur Form (BNF).	Linz: Chapter 6
Week 14	Equivalence of pushdown automata and context-free grammars. Construction of PDA from CFG and CFG from PDA. Pumping Lemma for context-free languages	Linz: Chapters 7–8 Quiz7
Week 15	Introduction to Turing machines (TM). Motivation and limitations of previous computational models. Formal definition of TM. Representation of TM using transition diagrams and transition functions. Design of Turing machines for simple problems. Turing machines with output.	Linz: Chapter 9 Quiz8
Week 16	Variants of Turing machines. Universal Turing machine. Introduction to decidability and undecidability. Applications of Turing machines in computational theory.	Linz: Chapters 10–12