Lesson Plan No. 1	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: Learn the foundational concepts of set theory, including unions, intersections, and complements, and apply these concepts in solving numerical problems.
Teaching Aids (if any)	a. Power point presentation b. OpenBoard
Teaching Development	1. **Introduction** (5 minutes): Brief overview of set theory and its importance 2. **Design Thinking** (30 minutes) Detailed discussion on the basics of sets, including definitions and notations. Introduction to operations on sets (union, intersection, difference, and complement) with examples. 3. **Exercise (5 minutes) –** Solve 2-3 basic set operation problems.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Video Lecture : "Basics of Set Theory" https://youtu.be/58N2N7zJGrQ?list=PLBlnK6fEyqRgp46KUv4ZY69yXmpwKOIev Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions on set operations. 2. Spend 5 minutes to evaluate understanding through problem-solving.

Lesson Plan No. 1.1	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept and structure of Deterministic Finite Automata (DFA). b. Design simple DFA for given patterns. c. Analyze the computational capabilities of DFA.
Teaching Aids (if any)	c. Power point presentation d. DFA simulation tools (like JFLAP).
Teaching Development	**1. Introduction** (5 minutes) • What is Automata Theory? Why is it important? • Definition and concept of DFA. • Components of DFA (states, alphabet, transition function, start state, accept states). **2. DFA concepts** (30 minutes) • Explaining the structure of DFA with examples. • Transition diagrams and tables. • Language recognition by DFA. • Examples of simple DFAs (e.g., recognizing patterns in strings). **3.Exercise (5 minutes)** – • Students design a DFA for a given pattern (e.g., strings ending with 'ab'). • Group activity: Construct DFA for different sets of strings.
Closure	1.Summarize DFA concepts. 2.Discuss the significance of DFA in computer science. 3.Review of DFA design exercise. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Short quiz on DFA concepts. 3. Analysis of the designed DFA by students. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	**Course Name: Theory of Computation**	**Course No.: COM-604**

Objectives	At the end of the lesson the student shall be able to: a.Understand the concept of Non-deterministic Finite Automata (NFA) without Epsilon moves. b.Design NFAs for various patterns and languages. c.Differentiate between DFA and NFA.
Teaching Aids (if any)	a.Power point presentation b.Interactive whiteboard for diagramming NFAs. c.NFA patterns and exercises. d.NFA simulation tools (like JFLAP).
Teaching Development	**1.Introduction** (5 minutes) • Brief revision of DFA. • Introduction to NFA and its distinction from DFA. • Basic components of NFA. **2.Design Thinking** (30 minutes) • Explaining NFA with examples. • Transition functions in NFA. • Understanding non-determinism in state transitions. • Examples of simple NFAs **3.Exercise (5 minutes) –** Students create NFAs for given patterns (e.g., strings containing '01'). Group discussion on converting simple patterns into NFA.
Closure	1.Comparison of DFA and NFA. 2.Discuss the power and limitations of NFA. 3.Recapitulate the exercise findings. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2.Quiz on NFA design and concepts. 3.Peer review of constructed NFAs. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the methods to convert DFA to NFA and vice versa. b.Apply conversion algorithms in practical examples. c.Analyze the implications of conversions on automata design.
Teaching Aids (if any)	a.Power point presentation b.Step-by-step conversion guide. c.Worksheets with conversion exercises. d.Conversion algorithm flowcharts.
Teaching Development	**1.Introduction** (5 minutes) • Review of DFA and NFA structures. • Why conversion is important? • Overview of conversion methods. **2.Conversion Techniques** (30 minutes) • Demonstrating DFA to NFA conversion. • NFA to DFA conversion using subset construction method. • Worked examples for each conversion. **3.Exercise (5 minutes) –** **Students convert provided DFA to NFA and vice versa.** **Small group discussions on conversion challenges.**
Closure	1.Review of conversion techniques. 2.Discuss the efficiency and practicality of conversions. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Conversion exercise evaluation. 2.Short quiz on conversion algorithms. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the concept of equivalence between DFA and NFA. b.Prove the equivalence for specific automata. c.Appreciate the theoretical importance of this equivalence.
Teaching Aids (if any)	a.Power point presentation b.Comparative diagrams of DFA and NFA. c.Case studies showing equivalence. d.Interactive quizzes.
Teaching Development	**1.Introduction** (5 minutes) • Recap of DFA and NFA basics. • Introduction to the concept of equivalence. **2.Equivalence Concepts** (30 minutes) • Detailed explanation of equivalence. • Proving equivalence through examples. • Language acceptance by both DFA and NFA. **3.Exercise (5 minutes)** – **Students work on proving equivalence between given DFA and NFA.** **Group discussions on different approaches to prove equivalence.**
Closure	1.Summarize the significance of equivalence. 2.Review key takeaways from the exercises Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Quiz on equivalence concepts. 2.Evaluation of equivalence proofs by students. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the concept and structure of NFA with Epsilon-moves. b.Design NFA with E-moves for specific language patterns. c.Discuss the implications of E-moves in automata
Teaching Aids (if any)	a.Power point presentation b.Visual diagrams of NFA with E-moves. c.Interactive exercises and examples. d.E-move elimination methods and examples.
Teaching Development	**1.Introduction** (5 minutes) • What are E-moves in NFA? • The role of Epsilon transitions in NFA. **2.Design Thinking** (30 minutes) • Structure and functioning of NFA with E-moves. • Designing NFA with E-moves. • E-move elimination technique. **3.Exercise (5 minutes) –** **Students design their own NFA with E-moves.** **Group activity to eliminate E-moves from given NFAs.**
Closure	1.Discussion on the utility and challenges of E-moves. 2.Recapitulation of E-move elimination techniques. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Quiz on NFA with E-moves. 2.Peer review of designed NFAs. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the fundamentals of regular expressions. b.Design regular expressions for specific patterns. c.Learn to convert regular expressions to NFAs.
Teaching Aids (if any)	a.Power point presentation b.Examples of regular expressions. c.Regular expression to NFA conversion guide. d.Practice problems for regular expression design.
Teaching Development	**1.Introduction** (5 minutes) Basics of regular expressions. Significance in pattern matching and automata. **2.Regular Expressions Concepts** (30 minutes) Syntax and semantics of regular expressions. Building blocks of regular expressions. Conversion of regular expressions to NFAs. **3.Exercise (5 minutes) –** Designing regular expressions for given patterns. Students convert their regular expressions to NFAs.
Closure	1.Review of regular expression concepts. 2.Discussion on practical applications. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Regular expression design exercise. 2.Quiz on regular expressions and their conversion to NFAs. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the concept of finite machines with output. b.Differentiate between automata with and without output functions. c.Learn to design finite machines with specific output behaviors.
Teaching Aids (if any)	a.Power point presentation b.Diagrams of finite machines with output. c.Case studies of finite machines applications. d.Design exercises for finite machines with outputs.
Teaching Development	**1.Introduction** (5 minutes) Overview of finite machines with output. Distinction between finite automata and output machines. **2.Finite Machine** (30 minutes) Structure and function of output in finite machines. Types of output functions (e.g., Mealy and Moore machines). Practical examples of output machines. **3.Exercise (5 minutes)** – **Students design a finite machine with a specified output.** **Discussion and analysis of student designs.**
Closure	1.Discuss the significance of output in finite machines. 2.Review of student-designed machines. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Design exercise assessment. 2.Quiz on finite machines with output. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the structures of Moore and Mealy machines. b.Compare and contrast Moore and Mealy machines. c.Design examples of both Moore and Mealy machines.
Teaching Aids (if any)	a.Power point presentation b.Comparative charts for Moore and Mealy machines. c.Diagrammatic examples. d.Design templates for both machine types.
Teaching Development	**1.Introduction** (5 minutes) Introduction to Moore and Mealy machines. Fundamental differences between the two. **2.Moore and Mealey Machine Concepts** (30 minutes) Detailed study of Moore machine structure and behavior. Examination of Mealy machine characteristics. Comparative analysis of both machines. **3.Exercise (5 minutes)** – **Students design a simple Moore and a Mealy machine.** **Group discussion on the design choices and differences.**
Closure	1.Summarizing key differences and similarities. 2.Discussing practical applications of each machine type. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Assessment of designed machines. 2.Quiz on Moore and Mealy machine concepts. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 9	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Learn the conversion techniques between Moore and Mealy machines. b.Understand the equivalence of these machines in representing functions. c.Apply conversion techniques in practical examples.
Teaching Aids (if any)	a.Power point presentation b.Step-by-step conversion guides. c.Comparative analysis worksheets. d.Example conversions for practice.
Teaching Development	**1.Introduction** (5 minutes) • Review of Moore and Mealy machines. • Why study conversion and equivalence? **2.Conversion and Equivalence Techniques** (30 minutes) • Methods for converting Moore to Mealy machines and vice versa. • Discussion on the equivalence of these conversions. • Practical examples of conversions. **3.Exercise (5 minutes) –** **Students perform conversions between Moore and Mealy machines.** **Group activity on analyzing conversion outcomes.**
Closure	1.Review of conversion methods. 2.Discussion on the practical significance of these conversions. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Conversion exercise review. 2.Quiz on conversion and equivalence concepts. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	Course Name: Theory of Computation	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a.Understand the Myhill-Nerode Theorem and its significance. b.Learn to apply the theorem in automata theory. c.Use the theorem for proving the minimality of automata.
Teaching Aids (if any)	a.Power point presentation b.Theoretical explanation of the Myhill-Nerode Theorem. c.Examples and applications in automata. d.Exercises for theorem application.
Teaching Development	**a.Introduction** (5 minutes) • Basic introduction to the Myhill-Nerode Theorem. • Its place and importance in automata theory. **b.Design Thinking** (30 minutes) • Detailed explanation of the theorem. • Application of the theorem in proving language regularity. • Using the theorem to discuss automata minimality. **c.Exercise (5 minutes) –** **Students apply the theorem to given automata examples.** **Group discussion on theorem application outcomes.**
Closure	1.Summarizing the theorem's implications in automata theory. 2.Review of exercises and applications. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1.Application-based exercise assessment. 2.Quiz on the Myhill-Nerode Theorem. Spend 5 minutes to evaluate student assimilation of the lesson contents