Department of MBA

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Theory of Computation
2.	Course Code	COM-503
3.	Academic Year	2024-2025
4.	Semester	Vth
5.	Number of Lesson plans	32
6.	Faculty Assigned	Anil Gupta

Faculty Signature

Lesson Plan No. 1	Course Name:Theory of Computation Topic Name:Introduction to Theory of Computation	Course No.: COM-503

Objectives	At the end of the lesson, the student shall be able to: 1. Understand the basic principles of the Theory of Computation. 2. Define and explain key terms such as automata, formal languages, and computational problems.
TeachingAids (if any)	a. PPT b. Video
TeachingDevelopment	**Introduction (5 minutes)** **Engagement**: • Use real-time examples of simple computational devices in daily life (e.g., traffic lights, vending machines) to introduce the concept of computation. • Mention that many of the devices and systems they use daily operate on principles from the Theory of Computation. **Presentation:** o Introduce the concept of the Theory of Computation. o Highlight the importance of understanding computational processes and the history of computation. o Introduce the concept with Mod-01 Lec-01 What is theory of computation? (youtube.com) **Development (30 minutes)** • **Basic Concepts in Theory of Computation:** o Define key terms: automata, formal languages, and computational problems. o Provide simple examples to illustrate these concepts. • **Types of Automata:** o Introduce different types of automata: finite automata, pushdown automata, and Turing machines. o Explain the differences and uses of each type of automaton. • **Formal Languages and Grammars:** o Define formal languages and explain their role in computation. o Introduce different types of grammars: regular, context-free,

	3.Exercise (5 minutes) • Provide different example problems for students to solve: o Identify the type of automaton suitable for a given language. o Create a simple grammar for a specific language. o Discuss how a real-world problem can be modeled using concepts from the Theory of Computation.
Closure	• Summarize key concepts learned about the basics of the Theory of Computation. 1. Homework - Solve the quiz questions on computational devices and model concepts and submit them on Google Classroom Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	• **Reflective Questions:** • Allow students to answer and discuss questions such as • "What is the Theory of Computation?", • "How do we classify different types of automata?", and "What are the applications of the Theory of Computation?". **Assessment:** • Conduct a Google Form quiz to evaluate student assimilation of the lesson contents. • Spend 5 minutes evaluating student assimilation of the lesson contents

Lesson Plan No. 2	Course Name:Theory of Computation Topic Name:Deterministic Finite Automata (DFA)	Course No.: COM-503

Objectives	At the end of the lesson, the student shall be able to: a. Understand the concept of a Deterministic Finite Automaton (DFA) and its components, including states, transitions, alphabet, and accepting states. b. Use a DFA to recognize whether a given input string belongs to a specified language. c. To represent a DFA using a transition diagram and a transition table.
TeachingAids (if any)	a. PPT b. Video
TeachingDevelopment	**Introduction (5 minutes)** • **Engagement:** • Use real-time examples of simple computational devices in daily life (e.g., traffic lights, vending machines) to introduce the concept of automata. • • Explain that many of the devices and systems they use daily operate on principles from the Theory of Computation. • **Presentation:** o Introduce the concept of a Deterministic Finite Automaton (DFA). o Highlight the importance and applications of DFAs in various fields. o Introduce the concept of DFA construction with o Lecture 01: Deterministic Finite Automata (DFA) (youtube.com) **Development (30 minutes)** • **Components of DFA:** o Define and explain the components of a DFA: states, alphabet, transitions, start state, and accepting states. o Provide examples to illustrate these components. • **Constructing a DFA:** o Explain the process of constructing a DFA. o Use examples to demonstrate how to create a DFA for a given language. o Show a video from NPTEL on DFA construction. • **Transition Diagrams and Tables:** o Explain how to represent a DFA using a transition diagram. o Illustrate how to create a transition table from a DFA. o Provide examples and practice problems.

	. **Exercise (5 minutes)** Provide different example problems for students to solve: • Design a DFA that accepts strings ending in '01'. • Create a DFA for the language of strings with an even number of zeros. • Represent these DFAs using transition diagrams and tables.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve quiz questions on DFAs and submit them on Google Classroom.
Evaluation	1.Reflective Questions Allow students to answer and discuss questions such as "What is a DFA?", "How do we construct a DFA?", and "What are the applications of DFAs?". 2.**Assessment:** Conduct a Google Form quiz to evaluate student assimilation of the lesson contents. student assimilation of the lesson

Lesson Plan No. 3	Course Name:Theory of Computation Topic Name:NFA without E moves	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Non deterministic finite automata b. Define the formal definition of NFA c. Implement the graphical representation of NFA d. Illustrate the differences between NFAand DFA
TeachingAids (if any)	a. Chalk and talk b. PPT
TeachingDevelopment	1. **Introduction**(5 minutes) - Explain the homework questions on DFA. - Introducing the concept of transition diagram and transition states for NFA. - Articulate the computation of NFA. - Implement the transition diagrams for a few examples. - Introduce the concept of nfa construction with https://www.youtube.com/watch?v=wgC3BCyjzbM&list=PL_Si_t2-nG7cMHfpLOZ88TgEPbXp489HZ&index=5 2. **Development**(30 minutes) a. Practise Problems - NFAwith ∑ = {0, 1} accepts all strings with 01. - Design an NFAwith ∑ = {0, 1} in which double '1' isfollowed by double '0'. - Design an NFAin which all the string contains a substring1110 - Introduce concept ofTransition state, transition functions,initial state, final state, set of states for NFA. b. TransitionTable - Illustrate the representation of transition table - Illustrate construction of transition tables for the given examples. 3. **Exercise** (5 minutes) – Solve the following for both transition table and diagram:

	- Design an NFAwith ∑ = {0, 1} accepts all string in which the third symbol from the right end is always 0. - Design a NFAwith ∑ = {0, 1} accepts the strings with an even number of 0's followed by single 1. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the NFA and submit on Googleclassroom Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Nearpod/ Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

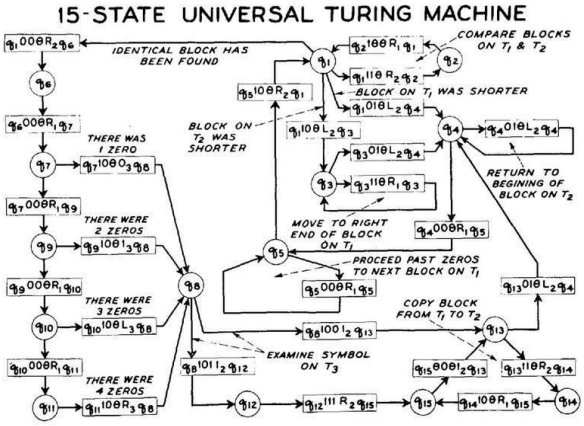
Lesson Plan No. 4	Course Name:Theory of Computation Topic Name:NFA with E-moves	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of Non deterministic finite automata b. Define the E transition on NFA c. Illustrate the elimination of E transition from NFA d. Articulate the E closure property e. Conversion from NFAto DFA
TeachingAids (if any)	a. Chalk and talk b. Video of NPTEL c. PPT
TeachingDevelopment	1. **Introduction**(5 minutes) - Explain the homework questions on NFA. - Introducing the concept of E transition for NFA. - Articulate the computation of NFAto DFAconversion. - Introduce the concept of NFAto DFAconstruction with https://www.youtube.com/watch?v=nyH7zyx7J5I&list=PL_Si_t2-nG7cMHfpLOZ88TgEPbXp489HZ&index=7 2. **Development**(30 minutes) a. Practise Problems - NFAwith ∑ = {0, 1} accepts all strings with 01. Convert itinto its equivalent DFA. - Design an NFAwith ∑ = {0, 1} in which double '1' isfollowed by double '0'. Convert it into its equivalent DFA. - Design an NFAin which all the string contain a substring1110. Convert it into its equivalent DFA. b. EquivalenceTheorem - Illustrate the theorem for proving the equivalence of NFAtoDFA. 3. **Exercise** (5 minutes) – Solve the following for both transition table and diagram : -Design an NFAwith ∑ = {0, 1} accepts all string in which thethird symbol from the right end is always 0.Convert it into

	- Design a NFAwith ∑ = {0, 1} accepts the strings with an evennumberof 0's followed by single 1.Convert it into its equivalentDFA. Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the DFAand submit on Googleclassroom Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Nearpod/ Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

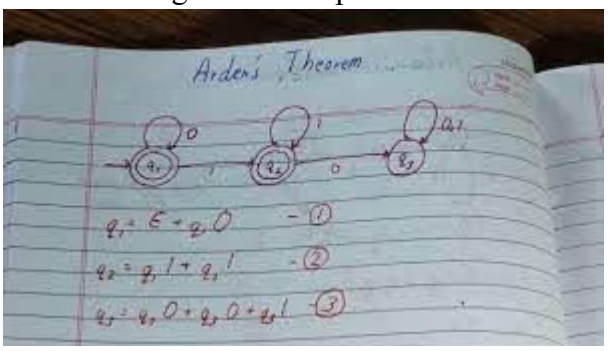
Lesson Plan No. 5	Course Name:Theory of Computation Topic Name:Mealy Mooore Machine	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Illustrate the composite machine conversion with the help of examples
TeachingAids (if any)	a. Chalk and talk b. Video of NPTEL c. PPT d. Use of Google Classroom tool for online quiz

TeachingDevelopment	1. **Introduction**(15 minutes) - Explain the homework questions on Moore Machine. - Introducing the steps of converting the Mealy to Moore Machine. - Articulate the Mealy to Moore Machine conversion with the help ofTransition Diagram andTransition table. - Introduce the concept of Mealy to Moore Machine conversion withhttps://www.youtube.com/watch?v=O3If0Nr9to0 2. **Development**(30 minutes) a. Elaborating the steps. - Design the composite conversion through the transition diagram and table.. b. Understanding with Example  3. **Exercise** (5 minutes) –

	Solve the following Mealy to Moorey Machine for transition diagram and table : - Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested solving more problems on Mealy Machine. - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the Moore Machine and submit on Google classroom Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	Course Name:Theory of Computation Topic Name:Arden's Theorem	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept ofArden's theorem b. Define the conversion process of arden's theorem c. Illustrate the conversion of regular expressions
TeachingAids (if any)	a. Chalk and talk b. PPT c. Use of Google Classroom tool for online quiz
TeachingDevelopment	1. **Introduction**(15 minutes) - Explain the homework questions on arden's theorem - Introducing the steps of converting the regular expressions - Articulate the composition problem with the help ofTransitionDiagram andTransition table. - Introduce the concept of arden's calculationwithhttps://www.youtube.com/watch?v=O3If0Nr9to0 2. **Development**(30 minutes) a. Elaborating the steps. - Design the composite conversion through the transition diagram and table.. b. Understanding with Example c.  3. **Exercise** (5 minutes) – -

	Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students to answer and discuss. 2. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

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

Objectives	At the end of the lesson the student shall be able to: a. Articulate the concept recursion. b. Understand problems with left and right recursion
TeachingAids (if any)	a. Chalk and talk b. PPT c. Use of Google Classroom tool for online quiz
TeachingDevelopment	1. **Introduction**(5 minutes) - Explain the homework questions onTuring machine. - Introducing the concept of RecursiveGrammar. - Explaining the concept of left and right recursion. - Articulate the concept of recursion. - Introduce the concept of recursion with https://youtu.be/tdBL2EEKgB8 2. **Development**(30 minutes) a. Practise Problems Consider the following grammar and eliminate left recursion- $E \rightarrow E + E / E \times E / a$ Consider the following grammar and eliminate left recursion- $A \rightarrow ABd / Aa / a$ $B \rightarrow Be / b$ 3. **Exercise** (5 minutes) – Consider the following grammar and eliminate left recursion- $S \rightarrow (L) / a$ $L \rightarrow L, S / S$ Consider the Left Recursion from the Grammar. $E \rightarrow E + T \mid T$ $T \rightarrow T^* F \mid F$ $F \rightarrow (E) \mid id$ Eliminate immediate left recursion from the Grammar

Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the recursion and submit on Google classroom Spend 5 minutes to wrap up and consolidate the leanings
Evaluation	1. Reflective Questions (What,Why,Who?). 2. Allow students to answer and discuss. 3. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

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

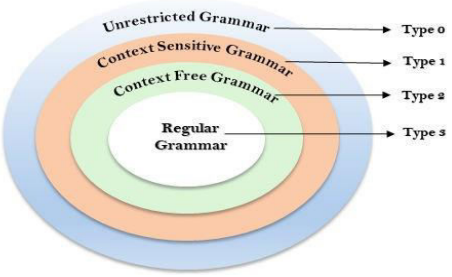
Objectives	At the end of the lesson the student shall be able to: a. Articulate the concept of ambiguity. b. Understanding left most and right most derivation of grammar. c. Demonstrate examples to understand ambiguity in grammar.
TeachingAids (if any)	a. Chalk and talk b. PPT c. Use of Google Classroom tool for online quiz

TeachingDevelopment	1. **Introduction**(5 minutes) - Explain the homework questions on RecursiveGrammar. - Introducing the concept of ambiguity. - Explaining the concept of left most and right most derivation of grammar. - Introduce the concept of ambiguity with https://youtu.be/rjWqEI3HOD4 2. **Development**(30 minutes) Consider a grammar G with the production rule 1. E → I 2. E → E + E 3. E → E * E 4. E → (E) 5. I → ε | 0 | 1 | 2 | ... | 9 And derive string 3*2+5 Check whether the given grammar G is ambiguous or not. 1. E → E + E 2. E → E - E 3. E → id 3. **Exercise** (5 minutes) – Check whether the given grammar G is ambiguous or not. 1. A→AA 2. A→ (A) 3. A→ a Consider the grammar- 1. S →A1B 2. A→ 0A/∈ 3. B → 0B / 1B /∈ For the string w = 00101 draw left and right most derivation Use Nearpod to collect responses and discuss the answers.

Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the left most and right most derivation and submit on Google classroom Spend 5 minutes to wrap up and consolidate the leanings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

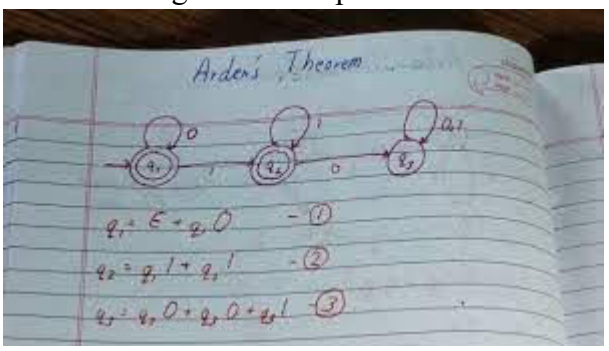
Lesson Plan No. 9	Course Name:Theory of Computation Topic Name:Normal Forms	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Articulate the concept of Chomsky Normal Form. b. Demonstrate outline of Normalization. c. Understand the CNF through examples.
TeachingAids (if any)	a. Chalk and talk b. PPT

TeachingDevelopment	1. **Introduction**(5 minutes) - Explain the homework questions on grammar ambiguity. - Articulate the concept of Chomsky Normal Form. - Demonstrate outline of Normalization. - Understand the CNF through examples.  - Introduce the concept of ambiguity with https://youtu.be/QG9hOwowaXI 2. **Development**(30 minutes) a. Practise Problems Find a grammar equivalent to S →AB/CA A→a B→BC/AB C→aB/b Convert the grammar with productions to Chomsky normal form S→ABa A→aab B→Ac **Exercise** (5 minutes) – Convert the grammar with productions to Chomsky normal form S→AB/aB A→aab/∈ B→bbA Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested Reading - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework - Solve the questions on the Chomsky normal form and submit on Google classroom Spend 5 minutes to wrap up and consolidate the leanings

Evaluation	1. Reflective Questions (What,Why,Who?).Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

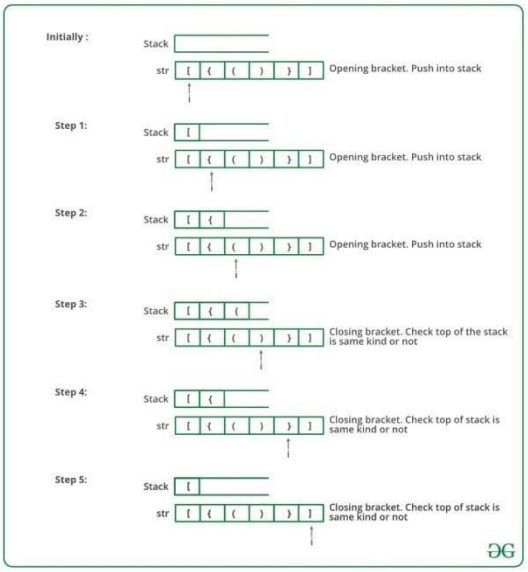
Lesson Plan No. 10	Course Name:Theory of Computation Topic Name:Arden's Theorem	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept ofArden's theorem b. Define the conversion process of arden's theorem c. Illustrate the conversion of regular expressions
TeachingAids (if any)	a. Chalk and talk b. PPT
TeachingDevelopment	1. **Introduction**(15 minutes) - Explain the homework questions on arden's theorem - Introducing the steps of converting the regular expressions - Articulate the composition problem with the help ofTransitionDiagram andTransition table. - Introduce the concept of arden's calculationwithhttps://www.youtube.com/watch?v=O3If0Nr9to0 2. **Development**(30 minutes) a. Elaborating the steps. - Design the composite conversion through the transition diagram and table.. b. Understanding with Example c.  3. **Exercise** (5 minutes) – -

	Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Nearpod/ Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

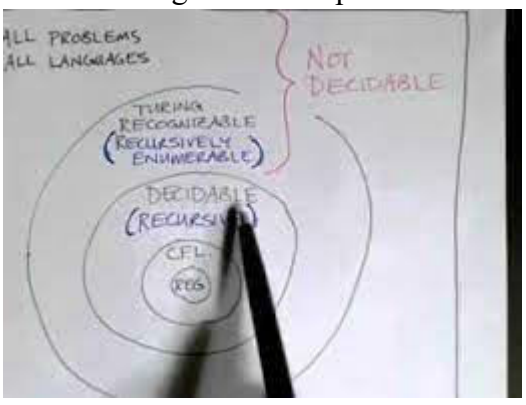
Lesson Plan No. 11	Course Name:Theory of Computation Topic Name:Application of Turing Machine	Course No.: COM-604

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of parenthesis checker by turing machines b. Define the parenthesis checker c. Illustrate the conversion of parenthesis checker
TeachingAids (if any)	a. Chalk and talk b. Video of NPTEL c. PPT d. Use of Nearpod or Google Classroom tool for online quiz

TeachingDevelopment	1. **Introduction**(15 minutes) - Explain the homework questions on parenthesis checker - Introducing the steps of parenthesis checker - Articulate the parenthesis checker - Diagram andTransition table. - Introduce the concept of parenthesis checker calculation withhttps://www.youtube.com/watch?v=O3If0Nr9to0 2. **Development**(30 minutes) a. Elaborating the steps. - Design the composite conversion through the transition diagram and table.. b. Understanding with Example c. 

	3. **Exercise** (5 minutes) – - Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Nearpod/ Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 12	Course Name:Theory of Computation Topic Name:Church's Hypothesis	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept of church's hypothesis b. Define the partial computable functions c. Illustrate the conversion of regular expressions
TeachingAids (if any)	a. Chalk and talk b. Video of NPTEL c. PPT d. Use of Nearpod or Google Classroom tool for online quiz
TeachingDevelopment	1. **Introduction**(15 minutes) - Explain the homework questions on church's hypothesis - Introducing the steps of computable functions - Articulate the church hypothesis evaluation - Diagram andTransition table. - Introduce the concept of arden's calculationwithhttps://www.youtube.com/watch?v=O3If0Nr9to0 2. **Development**(30 minutes) a. Elaborating the steps. - Design the composite conversion through the transition diagram and table.. b. Understanding with Example c.  3. **Exercise** (5 minutes) – -

	Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmationfrom students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 13	Course Name:Theory of Computation Topic Name:Church's Hypothesis	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: By the end of this lesson, students will be able to: a. Understand the concept of Turing Machines and their importance in computability theory.
TeachingAids (if any)	a. PPT b. Video of NPTEL

TeachingDevelopment	1. **Introduction**(15 minutes)
	
	• **Warm-Up Questions:**
	  ○ Give examples of computational devices in daily life?
	• **Introduction to Turing Machines:**
	  ○ Define Turing Machines and discuss their role in theoretical computer science.
	  ○ Present the formal definition of a Turing Machine using slides.
	• **Contextual Importance:**
	  ○ Explain the significance of Turing Machines in the context of computational models and formal languages.
	  ○ Highlight practical applications and the impact of Turing Machines on modern computing.
	
	2. **Development**(30 minutes)
	
	• **Concepts:**
	  ○ Define and illustrate computational devices, abstract machines, and computational models.
	  ○ Discuss alphabets, strings, and languages with relevant examples.
	• **Video:**
	  ○ Show an introductory video on Theory of Computation (e.g., TOC Introduction Video).
	
	Turing Machines (20 minutes):
	
	• **Finite Tape:**
	  ○ Describe the components and function of a Turing Machine's tape.
	  ○ Illustrate how an input string is processed by a Turing Machine with examples.
	•

	Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 14	Course Name:Theory of Computation Topic Name: Non-deterministic Turing	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: • Define Non-deterministic Turing Machines (NDTMs). • Understand how NDTMs differ from deterministic Turing Machines.
TeachingAids (if any)	• Slides/Presentation • Video on Non-deterministic Machines

TeachingDevelopment	1. **Introduction**(15 minutes)
	
	• **Warm-Up Questions:**
	  ○ Give examples of computational devices in daily life?
	• **Introduction to Turing Machines:**
	  ○ Define Turing Machines and discuss their role in theoretical computer science.
	  ○ Present the formal definition of a Turing Machine using slides.
	• **Contextual Importance:**
	  ○ Explain the significance of Turing Machines in the context of computational models and formal languages.
	  ○ Highlight practical applications and the impact of Turing Machines on modern computing.
	
	2. **Development**(30 minutes)
	
	• **Concepts:**
	  ○ Define and illustrate computational devices, abstract machines, and computational models.
	  ○ Discuss alphabets, strings, and languages with relevant examples.
	• **Video:**
	  ○ Show an introductory video on Theory of Computation (e.g., TOC Introduction Video).
	
	Turing Machines (20 minutes):
	
	• **Finite Tape:**
	  ○ Describe the components and function of a Turing Machine's tape.
	  ○ Illustrate how an input string is processed by a Turing Machine with examples.
	•

	Use Google Classroom to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested solving more problems on recursion - https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf 3. Homework Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What,Why,Who?).Allow students toanswer and discuss. 2. Google form Quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	Course Name:Theory of Computation Topic Name: Deterministic vs. Non-Deterministic Turing	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: 1. Compare deterministic and non-deterministic Turing Machines. 2. Understand their operational differences.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines
TeachingDevelopment	☐ **Introduction (15 minutes):** • **Warm-Up Questions:** o What is the difference between deterministic and non-deterministic processes? • **Introduction to DTMs and NDTMs:** o Define both types of Turing Machines. o Present formal definitions and operational differences using slides. ☐ **Development (30 minutes):** • **Concepts:** o Explain deterministic Turing Machines (DTMs) and their functioning. o Discuss non-deterministic Turing Machines (NDTMs) and their characteristics. • **Video:** o Show a video comparing DTMs and NDTMs. ☐ **Exercise (5 minutes):** • Compare examples of problems solved by DTMs and NDTMs. • Use Google Classroom to submit comparisons.

	Use Google Classroom to collect responses and discuss the answers.
Closure	☐ **Summarize Key Points:** • Recap the differences between DTMs and NDTMs. ☐ **Suggested Reading:** • Link: IITG Notes ☐ **Homework:** • Solve problems involving both DTMs and NDTMs.
Evaluation	☐ **Reflective Questions:** • What are the key differences between DTMs and NDTMs? ☐ **Quiz:** • Administer a Google Form quiz on DTMs and NDTMs.

Lesson Plan No. 16	Course Name:Theory of Computation Topic Name: Multitape Turing Machines.	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: • Define multitape Turing Machines. • Understand their functionality and advantages over single-tape Turing Machines.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines
TeachingDevelopment	**Teaching Development:** 1. **Introduction (15 minutes):** ○ **Warm-Up Questions:** ▪ How can having multiple tapes affect computation? ○ **Introduction to Multitape Turing Machines:** ▪ Define and explain the concept of multitape Turing Machines. ▪ Present formal definitions and examples using slides. 2. **Development (30 minutes):** ○ **Concepts:** ▪ Describe the components and operation of multitape Turing Machines. ▪ Illustrate with examples of multitape operations. ○ **Video:** ▪ Show a video on the advantages of multitape Turing Machines. 3. **Exercise (5 minutes):** ○ Solve examples involving multitape Turing Machines. ○ Use Google Classroom for responses and discussion.

Closure	• **Summarize Key Points:**
	  • Recap the benefits and functionality of multitape Turing Machines.
	• **Suggested Reading:**
	  • Link: IITG Notes
	• **Homework:**
	  • Write problems involving multitape Turing Machines

Evaluation	1. **Reflective Questions:**
	  ○ How do multitape Turing Machines improve computational efficiency?
	2. **Quiz:**
	  ○ Administer a Google Form quiz on multitape Turing Machines.

Lesson Plan No. 17	Course Name:Theory of Computation Topic Name: Universal Turing Machines	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: • Define Universal Turing Machines. • Understand their role and importance in computation.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines
TeachingDevelopment	**Teaching Development:** 1. **ntroduction (15 minutes):** ○ **Warm-Up Questions:** ▪ What is a universal machine? How is it useful? ○ **Introduction to Universal Turing Machines:** ▪ Define and explain Universal Turing Machines. ▪ Present formal definitions and examples using slides. 2. **Development (30 minutes):** ○ **Concepts:** ▪ Describe the operation and significance of Universal Turing Machines. ▪ Illustrate with examples of universal computations. ○ **Video:** ▪ Show a video explaining Universal Turing Machines. 3. **Exercise (5 minutes):** ○ Solve problems related to Universal Turing Machines. ○ Use Google Classroom for responses. -

Closure	• **Summarize Key Points:**
	  • Recap the functionality and importance of Universal Turing Machines.
	• **Suggested Reading:**
	  • Link: IITG Notes
	• **Homework:**
	  • Solve problems involving Universal Turing Machines.

Evaluation	• **Reflective Questions:**
	  • How does a Universal Turing Machine operate?
	• **Quiz:**
	  • Administer a Google Form quiz on Universal Turing Machines.
	  ○

Lesson Plan No. 18	Course Name:Theory of Computation Topic Name: Introduction to Pushdown Automata	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: 1. Define what a Pushdown Automaton (PDA) is. 2. Understand the basic components and functionality of a PDA.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines
TeachingDevelopment	**Teaching Development:** • **Introduction (10 minutes):** • **Warm-Up Question:** Discuss the limitations of finite automata. • Introduce PDA: Define PDA as an extension of finite automata with an additional stack for memory. • **Development (25 minutes):** • Define the components of a PDA: states, stack, input tape. • Present the formal definition of PDA: PDA=(Q,Σ,Γ,δ,q0,Z0,F)\text{PDA} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)PDA=(Q,Σ,Γ,δ,q0,Z0,F). • Use slides to illustrate examples and diagrams of PDA components and their functions. **Exercise (10 minutes):** • Draw a simple PDA and describe its components. • Submit responses and discuss via Google Classroom.

Closure	• Recap the definition and components of PDA. • **Homework:** Read the textbook sections on PDAs and their applications.
Evaluation	• **Reflective Question:** What role does the stack play in a PDA? • **Quiz:** Short quiz on the basic components and definitions of a PDA.

Lesson Plan No. 19	Course Name:Theory of Computation Topic Name: PDA Operation and Transitions	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: 1. Understand how transitions work in a PDA. 2. Describe the PDA operation with example transitions.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines
TeachingDevelopment	• **Introduction (10 minutes):** • **Warm-Up Question:** How do transitions in finite automata compare to those in PDAs? • Introduce the transition function δ\deltaδ. • **Development (25 minutes):** • Explain the transition function δ:Q×Σ×Γ→2(Q×Γ∗)\delta: Q \times \Sigma \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}δ:Q×Σ×Γ→2(Q×Γ∗). • Illustrate with examples how transitions are defined and executed. • Use slides to show diagrams of PDA transitions. • **Exercise (10 minutes):** • Create and analyze PDA transition diagrams. • Use Google Classroom for submissions.

Closure	• Recap PDA transitions and their importance. • **Homework:** Solve problems involving PDA transitions.
Evaluation	1. **Reflective Question:** How does the transition function δ affect PDA computations? 2. **Quiz:** Simple questions on PDA transitions and examples.

Lesson Plan No. 20	Course Name:Theory of Computation Topic Name: PDA Operation and Transitions	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: 1. Understand how transitions work in a PDA. 2. Describe the PDA operation with example transitions.
TeachingAids (if any)	1. Slides/Presentation 2.Video on Non-deterministic Machines

TeachingDevelopment	• **Introduction (10 minutes):**
	  • **Warm-Up Question:** How do transitions in finite automata compare to those in PDAs?
	  • Introduce the transition function δ\deltaδ.
	• **Development (25 minutes):**
	  • Explain the transition function δ:Q×Σ×Γ→2(Q×Γ∗)\delta: Q \times \Sigma \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}δ:Q×Σ×Γ→2(Q×Γ∗).
	  • Illustrate with examples how transitions are defined and executed.
	  • Use slides to show diagrams of PDA transitions.
	• **Exercise (10 minutes):**
	  • Create and analyze PDA transition diagrams.
	  • Use Google Classroom for submissions.

Closure	• Recap PDA transitions and their importance. • **Homework:** Solve problems involving PDA transitions.
Evaluation	1. **Reflective Question:** How does the transition function δ affect PDA computations? 2. **Quiz:** Simple questions on PDA transitions and examples.

Lesson Plan No. 21	Course Name:Theory of Computation Topic Name: PDA Acceptance by Empty Stack	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: PDA Acceptance by Final State
TeachingAids (if any)	Equivalence of Acceptance by Final State and Empty Stack
TeachingDevelopment	1. **Introduction (10 minutes):** ○ **Warm-Up Question:** Why is it important to understand the equivalence of different acceptance methods? ○ Introduce the concept of equivalence between acceptance methods. 2. **Development (25 minutes):** ○ Prove the equivalence between acceptance by final state and empty stack. ○ Provide examples and formal proofs using slides. 3. **Exercise (10 minutes):** ○ Given a PDA, demonstrate that it accepts the same language whether using final state or empty stack acceptance. ○ Use Google Classroom for submissions.

Closure	• Recap the equivalence of acceptance methods. • **Homework:** Prove equivalence for additional examples.
Evaluation	• **Reflective Question:** What are the implications of the equivalence between acceptance methods for PDA design? • **Quiz:** Questions on equivalence of acceptance methods.

Lesson Plan No. 22	Course Name:Theory of Computation Topic Name: PDA Acceptance by Empty Stack	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: • Review the definition and components of CFG. • Understand how CFGs are used in formal language theory.
TeachingAids (if any)	Equivalence of Acceptance by Final State and Empty Stack
TeachingDevelopment	1. **Introduction (10 minutes):** ○ **Warm-Up Question:** What are the main components of a Context-Free Grammar? ○ Review the definition of CFG. 2. **Development (25 minutes):** ○ Explain the components of CFG (V, Σ, R, S). ○ Provide examples of CFGs and their derivations using slides. .

Closure	1. **Exercise (10 minutes):** o Given a CFG, generate strings and show derivations. o Use Google Classroom for submissions.
Evaluation	• **Reflective Question:** How do CFGs differ from regular grammars in terms of expressive power? • **Quiz:** Questions on CFG definitions and examples.

Lesson Plan No. 23	Course Name:Theory of Computation Topic Name: PDA Acceptance by Empty Stack	Course No.: COM-503

Objectives	At the end of the lesson the student shall be able to: • Review the definition and components of CFG. • Understand how CFGs are used in formal language theory.
TeachingAids (if any)	Equivalence of Acceptance by Final State and Empty Stack
TeachingDevelopment	1. **Introduction (10 minutes):** ○ **Warm-Up Question:** What are the main components of a Context-Free Grammar? ○ Review the definition of CFG. 2. **Development (25 minutes):** ○ Explain the components of CFG (V, Σ, R, S). ○ Provide examples of CFGs and their derivations using slides. .

Closure	1. **Exercise (10 minutes):** o Given a CFG, generate strings and show derivations. o Use Google Classroom for submissions.
Evaluation	• **Reflective Question:** How do CFGs differ from regular grammars in terms of expressive power? • **Quiz:** Questions on CFG definitions and examples.

Lesson Plan No. 24	Course Name:Theory of Computation Topic Name: Introduction to Recursive Languages	Course No.: COM-503

Objectives	Define recursive languages and their properties. b. Explain the concept of decidability. c. Provide real-world examples of recursive languages.
TeachingAids (if any)	Slides/Presentation
TeachingDevelopment	1. **Introduction (10 minutes)** ○ Explain recursive languages as languages for which a Turing machine can decide membership in finite time for every input. ○ Discuss the importance of decidability and how it relates to recursive languages. 2. **Development (30 minutes)** ○ **Definition and Properties:** Define recursive languages and discuss their characteristics, such as guaranteed termination and correctness. ○ **Examples:** ▪ **Example:** An algorithm that sorts a list of numbers in ascending order. The problem is solvable for every possible list within a finite amount of time, making it a recursive language. ○ **Decidability:** Explain how recursive languages are a subset of decidable problems, emphasizing that these problems have a guaranteed solution.

Closure	1. **Exercise (5 minutes)** ○ Present a set of problems and ask students to identify if they are recursive. ○ **Example:** Determine if a given integer is a prime number
Evaluation	Summarize key points on recursive languages and their properties. **Suggested Reading:** Textbook Chapter on Recursive Languages. ○ **Homework:** Write a brief report on a real-world problem that is a recursive language. ○ **Evaluation:** Reflective questions on recursive languages and their applications in real-world scenarios.

Lesson Plan No. 25	Course Name:Theory of Computation Topic Name: Introduction to Recursively Enumerable Languages	Course No.: COM-503

Objectives	a. Define recursively enumerable languages. b. Explain the concept of recognizability.
TeachingAids (if any)	Slides/Presentation
TeachingDevelopment	1. **Introduction (10 minutes)** ○ Define recursively enumerable languages as languages for which there exists a Turing machine that will accept strings in the language and may not halt on strings not in the language. ○ Discuss recognizability and how it relates to recursively enumerable languages. 2. **Development (30 minutes)** ○ **Definition and Properties:** Define recursively enumerable languages and their characteristics, including non-termination for strings not in the language. ○ **Examples:** ▪ **Example:** A Turing machine that accepts valid Python programs. If the program is correct, the machine will accept it. If not, the machine might run forever. ○ **Recognizability:** Explain how recursively enumerable languages can be recognized but not necessarily decided. 3. **Exercise (5 minutes)** ○ Present problems and ask students to identify if they are recursively enumerable. ○ **Example:** Identify if a given string belongs to a language defined by a non-terminating regular expression.

Closure	○ Summarize key points on recursively enumerable languages and recognizability. ○ **Suggested Reading:** Textbook Chapter on Recursively Enumerable Languages.
Evaluation	Reflective questions on recognizability and its impact on computational theory. For example: "How does the concept of recognizability influence our understanding of algorithmic problems?"

Lesson Plan No. 26	Course Name:Theory of Computation Topic Name: Context-Sensitive Languages and Linear Bounded	Course No.: COM-503

Objectives	a. Define context-sensitive languages and their properties. b. Explain the concept of Linear Bounded Automata (LBA).
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define context-sensitive languages as languages where the rules can be sensitive to the context of non-terminal symbols.
	  • Explain Linear Bounded Automata (LBA) as a type of Turing machine that operates within space bounded by the input size.
	• **Development (30 minutes)**
	  • **Context-Sensitive Languages:** Define and discuss their properties and use cases.
	  • **Examples:**
	  ○ **Example:** Parsing natural languages like English where context-sensitive grammars are used for complex syntactic structures.
	  • **LBA:** Explain LBAs and their role in recognizing context-sensitive languages.
	.

Closure	○ Summarize key points on context-sensitive languages and Linear Bounded Automata. ○ **Suggested Reading:** Textbook Chapter on Context-Sensitive Languages and LBA. ○ **Homework:** Write a brief report on how context-sensitive languages are applied in real-world scenarios like language translation. ○ **Evaluation:** Reflective questions on context-sensitive languages and LBAs. For example: "What practical applications can you identify where context-sensitive languages are necessary?"
Evaluation	○ Present examples of context-sensitive grammars and ask students to identify if they can be recognized by an LBA. ○ **Example:** Write a context-sensitive grammar for a simple language and discuss its recognition by an LBA.

Lesson Plan No. 27	Course Name:Theory of Computation Topic Name: Decidability and Undecidability	Course No.: COM-503

Objectives	a. Define decidability and undecidability in computational theory. b. Explain the implications of these concepts.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define decidability and undecidability in computational theory.
	  • Explain the concept of problems that can and cannot be solved algorithmically.
	• **Development (30 minutes)**
	  • **Decidability:** Discuss problems that are decidable with algorithms that terminate with a solution.
	  • **Examples:**
	  ○ **Example:** Checking if a given number is a prime number.
	  • **Undecidability:** Discuss problems that are undecidable, such as the halting problem.
	  • **Examples:**
	  ○ **Example:** Determining if a given arbitrary program will halt or run forever.
	• **Exercise (5 minutes)**
	  • Present problems and ask students to determine if they are decidable or undecidable.
	  • **Example:** Decide if a given simple algorithm will terminate for all possible inputs.

Closure	• Summarize key points on decidability and undecidability. • **Suggested Reading:** Textbook Chapter on Decidability and Undecidability. • **Homework:** Analyze a problem to classify it as decidable or undecidable.
Evaluation	Reflective questions on the impact of decidability and undecidability. For example: "How do the concepts of decidability and undecidability influence the design of algorithms?"

Lesson Plan No. 28	Course Name:Theory of Computation Topic Name: Decidability and Undecidability	Course No.: COM-503

Objectives	a. Define decidability and undecidability in computational theory. b. Explain the implications of these concepts.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define decidability and undecidability in computational theory.
	  • Explain the concept of problems that can and cannot be solved algorithmically.
	• **Development (30 minutes)**
	  • **Decidability:** Discuss problems that are decidable with algorithms that terminate with a solution.
	  • **Examples:**
	  ○ **Example:** Checking if a given number is a prime number.
	  • **Undecidability:** Discuss problems that are undecidable, such as the halting problem.
	  • **Examples:**
	  ○ **Example:** Determining if a given arbitrary program will halt or run forever.
	• **Exercise (5 minutes)**
	  • Present problems and ask students to determine if they are decidable or undecidable.
	  • **Example:** Decide if a given simple algorithm will terminate for all possible inputs.

Closure	• Summarize key points on decidability and undecidability. • **Suggested Reading:** Textbook Chapter on Decidability and Undecidability. • **Homework:** Analyze a problem to classify it as decidable or undecidable.
Evaluation	Reflective questions on the impact of decidability and undecidability. For example: "How do the concepts of decidability and undecidability influence the design of algorithms?"

Lesson Plan No. 29	Course Name:Theory of Computation Topic Name: Post's Correspondence Problem (PCP)	Course No.: COM-503

Objectives	a.Define Post's Correspondence Problem. b. Explain its significance in computational theory.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define Post's Correspondence Problem (PCP) and its significance in computational theory.
	  • Explain the problem of finding a sequence that matches given patterns.
	• **Development (30 minutes)**
	  • **PCP Definition:** Describe the PCP and its properties.
	  • **Examples:**
	  ○ **Example:** Provide an instance of PCP where students need to find sequences that match given patterns.
	  • **Undecidability:** Discuss why PCP is undecidable and its implications.
	• **Exercise (5 minutes)**
	  • Present a PCP instance and ask students to determine if a solution can be found or if the problem is undecidable.
	  • **Example:** Solve a simplified PCP problem using provided patterns.

Closure	• Summarize key points on Post's Correspondence Problem and its undecidability. • **Suggested Reading:** Textbook Chapter on Post's Correspondence Problem.
Evaluation	• **Evaluation:** Reflective questions on PCP. For example: "What challenges arise from the undecidability of PCP in computational theory?"

Lesson Plan No. 30	Course Name:Theory of Computation Topic Name: Recursive vs. Recursively Enumerable Languages	Course No.: COM-503

Objectives	a. Compare recursive and recursively enumerable languages. b. Explain their differences and similarities.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define recursive and recursively enumerable languages, highlighting their differences and similarities.
	• **Development (30 minutes)**
	  • **Recursive Languages:** Discuss their properties, including guaranteed termination and decidability.
	  • **Recursively Enumerable Languages:** Discuss their properties, including recognizability but not necessarily decidability.
	  • **Examples:**
	  ○ **Recursive Languages Example:** Simple arithmetic problems that are always solvable.
	  ○ **Recursively Enumerable Languages Example:** Problems where a solution may not be guaranteed, such as the Halting Problem.
	• **Exercise (5 minutes)**
	  • Present problems and ask students to classify them as either recursive or recursively enumerable.
	  • **Example:** Analyze a given problem to determine if it is recursive or recursively enumerable.

Closure	• Summarize key points on recursive and recursively enumerable languages. • **Suggested Reading:** Textbook Chapter on Recursive and Recursively Enumerable Languages. • **Homework:** Compare and contrast recursive and recursively
Evaluation	• Reflective questions on the distinctions and applications of each type. For example: "In what scenarios would you choose to use a recursively enumerable language over a recursive one?"

Lesson Plan No. 31	Course Name:Theory of Computation Topic Name: Recursive vs. Recursively Enumerable Languages	Course No.: COM-503

Objectives	a. Compare recursive and recursively enumerable languages. b. Explain their differences and similarities.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Define recursive and recursively enumerable languages, highlighting their differences and similarities.
	• **Development (30 minutes)**
	  • **Recursive Languages:** Discuss their properties, including guaranteed termination and decidability.
	  • **Recursively Enumerable Languages:** Discuss their properties, including recognizability but not necessarily decidability.
	  • **Examples:**
	  ○ **Recursive Languages Example:** Simple arithmetic problems that are always solvable.
	  ○ **Recursively Enumerable Languages Example:** Problems where a solution may not be guaranteed, such as the Halting Problem.
	• **Exercise (5 minutes)**
	  • Present problems and ask students to classify them as either recursive or recursively enumerable.
	  • **Example:** Analyze a given problem to determine if it is recursive or recursively enumerable.

Closure	• Summarize key points on recursive and recursively enumerable languages. • **Suggested Reading:** Textbook Chapter on Recursive and Recursively Enumerable Languages. • **Homework:** Compare and contrast recursive and recursively
Evaluation	• Reflective questions on the distinctions and applications of each type. For example: "In what scenarios would you choose to use a recursively enumerable language over a recursive one?"

Lesson Plan No. 32	Course Name:Theory of Computation Topic Name: Advanced Topics in Recursive and Recursively Enumerable LanguagesLanguages	Course No.: COM-503

Objectives	a. Explore advanced topics and recent research related to recursive and recursively enumerable languages. b. Discuss ongoing challenges and research directions.
TeachingAids (if any)	Slides/Presentation

TeachingDevelopment	• **Introduction (10 minutes)**
	  • Introduce recent advancements and research in the field of recursive and recursively enumerable languages.
	• **Development (30 minutes)**
	  • **Advanced Topics:** Discuss recent research findings, challenges, and trends.
	  • **Examples:**
	  ○ **Example:** Latest research on optimization problems related to recursively enumerable languages.
	  ○ **Example:** Emerging computational models that extend classical theory.
	• **Exercise (5 minutes)**
	  • Provide recent research papers and ask students to summarize key findings related to recursive and recursively enumerable languages.
	  • **Example:** Analyze a current research article and discuss its implications.

Closure	• Summarize key points on advanced topics and research directions. • **Suggested Reading:** Recent research papers and articles on advanced topics in computational theory. • **Homework:** Write a review of a recent research paper related to recursive and recursively enumerable languages.
Evaluation	Reflective questions on ongoing research. For example: "What are the most exciting recent developments in recursive and recursively enumerable languages?"