Lovely Professional University, Punjab

Course Code	Course Title	Lectures	Tutorials	Practicals	Credits
CSE408	DESIGN AND ANALYSIS OF ALGORITHMS	3	0	0	3
Course Weightage	ATT: 5 CA: 25 MTT: 20 ETT: 50				

Course Focus	EMPLOYABILITY,SKILL DEVELOPMENT

Course Outcomes :Through this course students should be able to

CO1 :: Understand the basic techniques of analyzing the algorithms using space and time complexity, asymptotic notations

CO2 :: apply the various string matching algorithms

CO3 :: Analyze the divide and conquer algorithm design technique using various problems

CO4 :: Evaluate the various dynamic programming and greedy algorithm design technique to solve various problems

CO5 :: Apply the Approximation Algorithm to solve some classic problems and design technique.

CO6 :: Define intractability (NP-completeness) and understand to solve the optimization problems

	TextBooks (T)		
Sr No	**Title**	**Author**	**Publisher Name**
T-1	INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHM	ANANY LEVITIN	PEARSON
T-2	INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHM	ANANY LEVITIN	PEARSON

	Reference Books (R)		
Sr No	**Title**	**Author**	**Publisher Name**
R-1	INTRODUCTION TO ALGORITHMS	C.E. LEISERSON, R.L. RIVEST AND C. STEIN	THOMAS TELFORD LTD.
R-2	THE DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS	A.V.AHO, J.E. HOPCROFT AND J.D.ULLMAN	PEARSON
R-3	COMPUTER ALGORITHMS - INTRODUCTION TO DESIGN AND ANALYSIS	SARA BAASE AND ALLEN VAN GELDER	PEARSON
R-4	FUNDAMENTALS OF COMPUTER ALGORITHMS	HOROWITZ, S. SAHNI	GALGOTIA PUBLICATIONS

Relevant Websites (RW)		
Sr No	**(Web address) (only if relevant to the course)**	**Salient Features**
RW-1	https://www.coursera.org/learn/analysis-of-algorithms	Foundations of Algorithm
RW-2	https://www.coursera.org/learn/algorithms-on-strings	String and its Matching Algorithms
RW-3	www.coursera.org/learn/linear-programming-and-approximation-algorithms	Introduction to Approximation Algorithms
RW-4	https://www.coursera.org/learn/dynamic-programming-greedy-algorithms	Divide and Conquer,Dynamic Programming ,Greedy Algoritms, NP completeness

LTP week distribution: (LTP Weeks)	
Weeks before MTE	7
Weeks After MTE	7
Spill Over (Lecture)	7

Detailed Plan For Lectures

Week Number	Lecture Number	Broad Topic(Sub Topic)	Chapters/Sections of Text/reference books	Other Readings, Relevant Websites, Audio Visual Aids, software and Virtual Labs	Lecture Description	Learning Outcomes	Pedagogical Tool Demonstration/ Case Study / Images / animation / ppt etc. Planned	Live Examples
Week 1	Lecture 1	Foundations of Algorithm (Analysis of algorithm: History and Motivation, A Scientific Approach, Example: Quicksort)	R-3	RW-1	Zero lecture for the Introduction of course ,objectives, structure sand details of academic tasks Introductionof Algorithm Scientific approach Basic knowledge about concepts of complexities of algorithms, its definition.	Student will learn the need of algorithms for real world applications	Demonstration with power point presentation	Solution to given problem

Week 1	Lecture 2	Foundations of Algorithm (Analysis of algorithm: History and Motivation, A Scientific Approach, Example: Quicksort)	R-3	RW-1	Zero lecture for the Introduction of course ,objectives, structure sand details of academic tasks Introductionof Algorithm Scientific approach Basic knowledge about concepts of complexities of algorithms, its definition.	Student will learn the need of algorithms for real world applications	Demonstration with power point presentation	Solution to given problem
	Lecture 3	Foundations of Algorithm (Introductions to "big-oh" notation and asymptotic analysis)		RW-1	Introductions to "big-oh" notation and asymptotic analysis. Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Merge sort , Master Theorem	Students will learn for asymptotic analysis and Types of Recurrences, Master Theorem	Demonstration with power point presentation and through Coursera	NA
Week 2	Lecture 4	Foundations of Algorithm (Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Mergesort , Master Theorem)	T-1	RW-1	Introductions to "big-oh" notation and asymptotic analysis. Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Merge sort , Master Theorem	Students will learn for asymptotic analysis and Types of Recurrences, Master Theorem	Demonstration with power point presentation and through Coursera	NA
	Lecture 5	Foundations of Algorithm (Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Mergesort , Master Theorem)	T-1	RW-1	Introductions to "big-oh" notation and asymptotic analysis. Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Merge sort , Master Theorem	Students will learn for asymptotic analysis and Types of Recurrences, Master Theorem	Demonstration with power point presentation and through Coursera	NA

Week 2	Lecture 6	Foundations of Algorithm (Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.)		RW-1	Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.	Student will learn the concepts of generating Functions	Self Learning through Coursera and discussion in classroom teaching	na
Week 3	Lecture 7	Foundations of Algorithm (Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.)		RW-1	Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.	Student will learn the concepts of generating Functions	Self Learning through Coursera and discussion in classroom teaching	na
	Lecture 8	Foundations of Algorithm (Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.)		RW-1	Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotic of Finite Sums, Asymptotic of Finite Sums.	Student will learn the concepts of generating Functions	Self Learning through Coursera and discussion in classroom teaching	na
	Lecture 9	Foundations of Algorithm (Trees: Trees and Forests, Binary Search Trees, Path Length, Other Types of Trees.)	T-2	RW-1	Trees: Trees and Forests, Binary Search Trees, Path Length, Other Types of Trees.	Students will learn the concepts of Non linear Data Structure Trees	Demonstration with power point presentation.	na

Week 4	Lecture 10	Foundations of Algorithm (Trees: Trees and Forests, Binary Search Trees, Path Length, Other Types of Trees.)	T-2	RW-1	Trees: Trees and Forests, Binary Search Trees, Path Length, Other Types of Trees.	Students will learn the concepts of Non linear Data Structure Trees	Demonstration with power point presentation.	na
	Lecture 11	String and its Matching Algorithms(Strings and Tries: Bit strings with Restrictions , Languages, Tries)		RW-1 RW-2	Strings and Tries: Bit strings with Restrictions , Languages, Tries	Students will learn the concepts of strings	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 12	String and its Matching Algorithms(Strings and Tries: Bit strings with Restrictions , Languages, Tries)		RW-1 RW-2	Strings and Tries: Bit strings with Restrictions , Languages, Tries	Students will learn the concepts of strings	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
Week 5	Lecture 13	String and its Matching Algorithms(Trie Parameters Key pattern matching concepts: Suffix Tree , Suffix Array, Knuth-Morris-Pratt algorithm)		RW-1 RW-2	Trie Parameters Key pattern matching concepts: Suffix Tree , Suffix Array, Knuth-Morris-Pratt algorithm	Students will learn String matching algorithm	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 14	String and its Matching Algorithms(Trie Parameters Key pattern matching concepts: Suffix Tree , Suffix Array, Knuth-Morris-Pratt algorithm)		RW-1 RW-2	Trie Parameters Key pattern matching concepts: Suffix Tree , Suffix Array, Knuth-Morris-Pratt algorithm	Students will learn String matching algorithm	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 15				Test 1			

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Week 6	Lecture 16	Divide and Conquer Technique(What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT)		RW-4	What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers ,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT	students will learn divide and conquer algorithms as a design scheme and learn some divide and conquer algorithms for Integer Multiplication (Karatsuba’s Algorithm), Matrix Multiplication (Strassen’s Algorithm), Fast Fourier Transforms (FFTs), and Finding Closest Pair of Points.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 17	Divide and Conquer Technique(What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT)		RW-4	What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers ,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT	students will learn divide and conquer algorithms as a design scheme and learn some divide and conquer algorithms for Integer Multiplication (Karatsuba’s Algorithm), Matrix Multiplication (Strassen’s Algorithm), Fast Fourier Transforms (FFTs), and Finding Closest Pair of Points.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem

Week 6	Lecture 18	Divide and Conquer Technique(What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT)		RW-4	What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer, Karatsuba’s Multiplication Algorithm , FFT Part 1: Introduction and Complex Numbers ,FFT Definition and Interpretation of Discrete Fourier Transforms, FFT: Divide and Conquer Algorithm for FFT, Application # 1 : Fast Polynomial Multiplication using FFT ,Application # 2: Data Analysis using FFT	students will learn divide and conquer algorithms as a design scheme and learn some divide and conquer algorithms for Integer Multiplication (Karatsuba’s Algorithm), Matrix Multiplication (Strassen’s Algorithm), Fast Fourier Transforms (FFTs), and Finding Closest Pair of Points.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
Week 7	Lecture 19				Test 2			
		SPILL OVER						
Week 7	Lecture 20				Spill Over			
	Lecture 21				Spill Over			
		MID-TERM						
Week 8	Lecture 22	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	

Week 8	Lecture 23	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 24	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	

Week 9	Lecture 25	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 26	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	

Week 9	Lecture 27	Dynamic Programming and Greedy Techniques (Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality)		RW-4	Introduction to Dynamic Programming + Rod Cutting Problem, Coin Changing Problem, Knapsack Problem, When Optimal Substructure Fails, Dynamic Programming: Longest Common Subsequence, Memorization, Coin Changing Problem. Introduction to Greedy Algorithms ,Greedy Interval Scheduling, Prefix Codes, Huffman Codes, Huffman Codes: Proof of Optimality	Students will learn about dynamic programming as a design principle for algorithms with a approach to formulating a problem as a dynamic program and solving these problems using memoization and will cover dynamic programming for finding longest common subsequences, Knapsack problem and some interesting dynamic programming applications.	Self Learning through Coursera and discussion in classroom teaching	
Week 10	Lecture 28	Approximation Algorithms (Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling)	R-3	RW-3	Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 29	Approximation Algorithms (Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling)	R-3	RW-3	Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 30	Approximation Algorithms (Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling)	R-3	RW-3	Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Week 11	Lecture 31	Approximation Algorithms (Approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem)	R-4	RW-3	approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 32	Approximation Algorithms (Approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem)	R-4	RW-3	approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	
	Lecture 33	Approximation Algorithms (Approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem)	R-4	RW-3	approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	
Week 12	Lecture 34	Approximation Algorithms (Travelling Salesman Problem and Approximation Schemes: Introduction to TSP and its applications, NP-Hardness of TSPs, Hardness of Approximating General TSPs)	R-1	RW-3	Travelling Salesman Problem and Approximation Schemes: Introduction to TSP and its applications, NP-Hardness of TSPs, Hardness of Approximating General TSPs	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 35	Approximation Algorithms (Travelling Salesman Problem and Approximation Schemes: Introduction to TSP and its applications, NP-Hardness of TSPs, Hardness of Approximating General TSPs)	R-1	RW-3	Travelling Salesman Problem and Approximation Schemes: Introduction to TSP and its applications, NP-Hardness of TSPs, Hardness of Approximating General TSPs	Students will learn the Approximation Algorithms	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem

Week 12	Lecture 36	Introduction to intractability (NP-completeness) and solving optimization problems.(Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm)	R-2	RW-4	Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm	students will learn about NP-completeness) and ow to solve optimization problems.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
Week 13	Lecture 37	Introduction to intractability (NP-completeness) and solving optimization problems.(Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm)	R-2	RW-4	Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm	students will learn about NP-completeness) and ow to solve optimization problems.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
	Lecture 38	Introduction to intractability (NP-completeness) and solving optimization problems.(Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm)	R-2	RW-4	Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm	students will learn about NP-completeness) and ow to solve optimization problems.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Week 13	Lecture 39	Introduction to intractability (NP-completeness) and solving optimization problems.(Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm)	R-2	RW-4	Decision Problems and Languages, Polynomial Time Problems, NP Definition, NP Completeness and Reductions ,NP Complete Problems: Examples, Computation and Physics ,Qubits and Operations, Bell's Inequality, Grover's Search Algorithm	students will learn about NP-completeness) and ow to solve optimization problems.	Self Learning through Coursera and discussion in classroom teaching	Solution to given problem
Week 14	Lecture 40				Test 3			
		SPILL OVER						
Week 14	Lecture 41				Spill Over			
	Lecture 42				Spill Over			
Week 15	Lecture 43				Spill Over			
	Lecture 44				Spill Over			
	Lecture 45				Spill Over			

Scheme for CA:

CA Category of this Course Code is:A0203 (2 best out of 3)

Component	Weightage (%)	Mapped CO(s)
Test 1	50	CO1, CO2
Test 2	50	CO3
Test 3	50	CO4, CO5, CO6

Details of Academic Task(s)

Academic Task	Objective	Detail of Academic Task	Nature of Academic Task (group/individuals)	Academic Task Mode	Marks	Allottment / submission Week
Test 1	To check the learning level	MCQ Based Class Test	Individual	Online	30	5 / 5

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Test 2	To Check the Learning level	MCQ based Class Test	Individual	Online	30	7 / 7
Test 3	To check the learning level	MCQ Based Class Test	Individual	Online	30	13 / 14

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.