GENERAL GUIDELINES

Do's:-

- Students should be on time for every lecture.
- Students are advised to show due respect to all faculty members.
- Students should keep the Classrooms, Laboratories and Workshops clean and tidy.
- Students must maintain absolute discipline and decorum, while on campus.
- **Students should come prepared with algorithm / flowchart / program / procedure for all the experiments before attending the laboratory session.**
- Students should bring the data sheets and laboratory records completed in all respects to the laboratory.
- Students are advised to clarify their doubts in the respective courses with the faculty.
- Students have to inform their parents that they should follow up the progress of their wards by being in touch with the institution authorities at regular intervals.
- **Students are advised to be present for the mentor meetings conducted by their respective Faculty Advisors, failing which appropriate disciplinary action will be taken.**

Don'ts:-

- Students are not permitted to attend the class without the identity card, once issued.
- Ragging is strictly prohibited because it is punishable under Karnataka Education Act. Any student involved in ragging, will be severely punished – which includes handing over the case to Police, rustication from the college etc.
- Writing on desks and walls is strictly prohibited, failing which the students will be fined heavily. If the identity of the individual is not established the entire class / students in the block will be fined.
- Students must not use their cell phones during class hours. If any student is found using their cell phone during class hours it will be confiscated.
- Students are not supposed to alter the configuration of the system / any software on the systems.

III SEMESTER (2021-25 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE21CS251A	Digital Design and Computer Organization	4	0	2	5	5	Icarus, Verilog Simulator, GTKWave waveform viewer	CC- Lab Integrated
2	UE21CS252A	Data Structures and its Applications	4	0	2	5	5	C Programming Language	CC-Lab Integrated
3	UE21CS241A	Statistics for Data Science	4	0	1	4	4	Python	CC-Independent
4	UE21CS242A	Web Technologies	4	0	1	4	4	HTML, CSS, JavaScript, MERN Technologies	CC-Independent
5	UE21CS243A	Automata Formal Languages and Logic	4	0	1	4	4	JFLAP	CC-Independent
6	UE21MA221A	Bridge Course Mathematics –I (Applicable for the Lateral Entry Students)*	2	0	0	2	0		FC-Independent
Total			20/22	0	7	22/ 24	22		
* - Audit Course									

UE21CS251A: DIGITAL DESIGN AND COMPUTER ORGANIZATION

No. of Credits: 5 **# of Hours: 100**

Class No.	Chapter title/ Reference literature	Topics to be covered	% of Portions covered	
			Absolute %	**Cumulative %**
UNIT – I: Combinational Logic Design				
1-2	**Lecture 1 slides**	Introduction, Boolean Functions, Truth Tables	21.5	21.5%
3-6	**R2:** 3.2-3.3 ,3.5-3.6	K-Maps		
7-8	**T1: Chapter 5 Digital Building Blocks,**5.1, 5.2.1	Adder/Subtractor, Overflow		
9-11: Lab1a	Basics of Verilog: Basic gates, Adder/ Subtractor (One bit & n-bit).			
12-14: Lab1b	Basics of System C: Basic gates, Adder/ Subtractor (One bit & n-bit).			
15-17	**T1: Chapter 2 Combinational Logic Design** 2.1, 2.2, 2.3,2.7	15-17		
18	**T1: Chapter 5 Digital Building Blocks** 5.2.5	18		
19-20: A1	Assignment-1: Write a Verilog code and test bench for Mux, Demux and Decoders.			
21	Revision			
UNIT – II: Combinational and Sequential Logic Design				

22	**T1: Chapter 2 Combinational Logic Design** 2.8, 2.6, Handouts 2.9 (exclude contamination delay)	Gate/Wire Delays, Timing	21.5	43%
24-26	**T1: Chapter 3 Sequential Logic Design** 3.2 (excluding 3.2.7) 3.3 (excluding 3.3.1, 3.3.3) 3.4 (excluding 3.4.4)	Latches, Flip-Flops		
27-30		Synchronous Logic Design		
31-33		Finite State Machines		
34-35: A2	Write a Verilog code and test bench for Flip-Flop, Synchronous Register			
36-38: Lab2	Design of an ALU			
39	Revision			
UNIT – III: Sequential Logic and Arithmetic Circuits				
40-43	**T1: Chapter 3 Sequential Logic Design** 3.4.1, examples 3.6, 3.7, 3.9	FSM examples	17.7	60.7%
44-45	**Handout** Link 1	Counters		
46-48: Lab3	Design of a Register File			
49	**T1: Chapter 5 Digital Building Blocks** 5.5, 5.5.1 5. 2.1	Memory Arrays		
50-52		Carry-lookahead and Prefix Adders		
53-54: A3	Write a Verilog code and test bench for Counters or Prefix adders.			
55-56: Lab4	Design of a Data-path (Integration of ALU and Register File).			
57	Revision			
UNIT – IV: Arithmetic Circuits and Architecture				
58-61	**R3: Chapter 9 Arithmetic** 9.4	Shift/add Multiplier/Divider	17.8	78.5%
62	**Handout** Link 2 29.2.3, 29.3.2	Wallace Tree Multiplier		
63-65: Lab5	Design of a Program Counter.			
66	**T1: Chapter 5 Digital Building Blocks** 5.3.2 (excluding subsections Rounding and Floating-Point Addition)	Floating point		
67-68	**Chapter 6 Architecture** 6.1, 6.2 6.3, 6.4.1 6.4.2, 6.4.3 (exclude switch/case statements) 6.4.4 (exclude magnitude comparison)	Introduction, Assembly Language		
69-70		Machine Language		
71-72: A4	Write a Verilog code and test bench for Multiplier or Divider circuit.			
73	Revision			
UNIT – V: Microarchitecture				

74-75	**T1: Chapter 6 Architecture** 6.5	Addressing modes	21.5	100%
76	**T1: Chapter 7 Microarchitecture** 7.1, 7.2, 7.4 (exclude 7.3.3, 7.3.4, 7.4.3)	Introduction, Performance Analysis		
77-81		Single-Cycle, Multi-Cycle Processor		
82	**Handout** Link 3	Systolic array matrix multiply		
83-85	**Handout** Link 4	Overview of Computer Systems Organization		
86-88: Lab6a	Design of a Control Logic.			
89-90: A5	Write a Verilog code and test bench to implement Microprocessor to execute Load and Jump Instructions			
91-92: Lab6b	Design of 16-bit Microprocessor			
93	Revision			
94-100:	Mini Project			

Textbook(s):

1) Digital Design & Computer Architecture, David Money Harris, Sarah L Harris

References:

1) Computer Organization and Design, David A Patterson, John L Hennessey

2) Digital Design, M.Morris Mano & Michael D. Ciletti

3) Computer Organization, Carl Hamacher, SafwatZaky, ZvonkoVranesic

DDCO LAB

Learning Outcome:

At the end of all lab experiment, the student will be able to:

1. Achieve knowledge and awareness of various components to design stable digital circuits.
2. Analyze and design combinational circuits.
3. Design and develop sequential circuits.
4. Design and develop a basic microprocessor.
5. Translate real world problems into digital logic formulations using Verilog.

Lab Policies:

1. You will (have access to and) work in the lab.
2. All communication will be done through Group Email. So, please keep checking emails for notifications and updates. Important information will also be emailed.
3. 15-minute lab discussion sessions will be held before the lab experiment. These will be conducted by the Lab in charge Faculty. It is advisable to read Theory Class notes before coming to the lab class, so that you are better prepared to ask questions and resolve doubts.
4. All Lab experiments are to be done individually.
5. Grading will occur in two parts: submission and demo.
6. For submission, upload all relevant files (specified with each lab under the 'Submission Details' section) via Edmodo/Google Drive. Once the lab is submitted, DO NOT make changes! You must demo with the code you submitted. In the event you decide to change the code for the demo, the day of the demo will be considered the turn-in date, and the appropriate late penalty will be applied.
7. In case you miss your check out lab slot, you can check out for that lab during office hours any time before the next lab's due date. In other words, the Faculty will not entertain requests for checking out labs older than the previous lab.
8. Late submissions will lead to penalty according to the following rules:
    a. One day late submission – less 10% of your normal score
    b. Two day late submission – less 20% of your normal score
    c. Three day late submission – less 30% of your normal score Submissions late by more than 3 days will not be accepted and you will be marked zero (unless you have taken permission from the professor).
    d. Sundays are not counted for late submissions. So, if a lab is due on Saturday and you submit it on Monday, it will be considered 1-day late submission.

UE21CS252A – Data Structures & its Applications (4-0-2-5-5)

# of Hours: 105

Class #	Unit#/ Reference Books	Topics to be Covered			% of Portion covered	
Color Code		Practice problems	Lab Exercises	Assignments	% of Syllabus	Cumulative %
1	Unit#1 Overview / T1, R1	Overview of the course, Introduction to Data Structures, User defined data type			22.85	22.85
2		Pointers, Pointer to structures				
3		Recursion				
4		Practice problems on C Fundamentals and Recursion				
5		Static and Dynamic Memory Allocation				
6		Practice problems on Files and Dynamic Memory Allocation				
7		Abstract Data Type (ADT), List as an ADT, List as a data structure				
8		Array List Implementation				
9		Singly Linked List (SLL) insert operations: beginning, end, at a specified position, destroy list operation				
10		SLL delete operations: beginning, end, at a specified position, search operation, concatenate list2 to list1				
11		Addition of two polynomials using SLL				
12		Delete every alternate node in the SLL starting from the first node				
13		Doubly Linked List (DLL) insert operations: beginning, end, at a specified position, destroy list operation				
14		DLL delete operations: beginning, end, at a specified position, search operation				
15		Create an ordered doubly linked list				
16		Merge two ordered lists				
17		Implementation of Circular Singly Linked List				
18		Implementation of Circular Doubly Linked List				
19		Sparse matrix and its representation using Multi list				
20		Skip list Case study				
21		Dictionary Implementation using skip list				
22		Revision / ISA1				
23-24		Assignment 1				

25	**Unit#2 Stacks and Queues / T1, R1**	Basic structure of a stack, Implementation of stack using arrays	20.95	43.80
26		Implementation of stack using linked list, Applications of stack: Function execution, Nested functions		
27		Evaluation of a postfix expression using stack		
28		Conversion of an expression from Infix to postfix using stack		
29-30		Conversion of an expression from Infix to prefix using stack and evaluate the prefix expression		
31		Parenthesis matching using stack		
32		Practice problems on stack and its applications		
33		Basic structure of a simple queue, Implementation of simple queue using array and linked list		
34		Implementation of circular queue using array and linked list		
35-36		Implement a queue using two stacks		
37		Implementation of priority queue using array		
38		Implementation of priority queue using linked list		
39		Double ended queue (Deque) and its Implementation using array		
40		Double ended queue (Deque) and its Implementation using linked list		
41		Applications of Queue: Case Study – Josephus problem, Implementation of Josephus problem		
42		Applications of Queue : CPU scheduling		
43		Practice problems on queue and its applications		
44		Revision / ISA2		
45-46		Assignment 2		
47	**Unit #3 Trees and Heaps / T1, R1**	Binary Tree and Binary Search Tree (BST) : definition, properties	19.05	62.85
48		Implementation of BST using dynamic allocation		
49		Binary Tree Traversal: Inorder, preorder, postorder		
50		Implementation of BST using arrays		
51		Binary Search Tree: node deletion operation		
52		Implementation of binary expression tree		
53-54		Implementation of the Iterative inorder, preorder, postorder traversal of a binary tree		
55		Threaded binary search tree and its implementation		
56		Implementation of Dictionary using Binary Search Tree		

57		Practice problems on binary trees		
58		n-ary tree, Forest, conversion of an n-ary tree and Forest to Binary Tree		
59		Heap and its properties, implementation using array, Top down heap construction		
60-61		Bottom up heap construction		
62		Implementation of Priority Queue using min and max heap		
63		Practice Problems on heap		
64		Revision / ISA3		
65-66		Assignment 3		
67	**Unit#4** **Balanced Trees and Graphs / T1, R1**	Balanced Trees: Definition, AVL trees	20	82.85
68		Rotations in AVL Trees		
69		Splay Trees		
70		Graphs: Introduction, properties, representation		
71		Implementation of graphs using adjacency matrix and adjacency list		
72		Depth First Search (DFS) traversal of a graph		
73		Breadth first search (BFS) traversal of a graph		
74		Application of BFS and DFS: Connectivity of graph		
75		Application of BFS and DFS: finding path in a network		
76-77		To check if there exists a cycle in a given graph		
78		Practice problems on graphs		
79		Representation of computer network topology using graphs		
80		Case Study: Indexing in databases (B Tree: K-way tree), B tree Insertion		
81		B Tree deletion operations with examples, Implementation for B Tree construction from given keys		
82-83		To find shortest path from source to destination		
84		Practice problems on graphs		
85		Revision / ISA4		
86-87		Assignment 4		
88	**Unit #5** **Suffix Tree and Hashing / T1, R1**	Hashing: Simple mapping, hash function, hash table	17.15	100
89		Collision handling using linear and quadratic probing		
90		Collision handling using separate chaining, Collision handling using double hashing and rehashing		
91-92		Implement collision handling using separate chaining and double hashing		

93		Practice Problems on hashing		
94		Implementation of Trie trees: Insert, search operations		
95		Trie: delete operation		
96-97		Display the words in a trie in lexicographic order		
98		Trie: multi pattern search, Suffix Trees, Applications of TRIE : URLs decoding, Word prediction using TRIE trees / Suffix Trees		
99		Auto-complete feature using Trie		
100		Revision / ISA5		
101		Practice problems on trie		
102-105		Assignment 5		

Tool/ Languages: C

Text Book:

1. “Data Structures using C / C++” , Langsum Yedidyah, Moshe J Augenstein, Aaron M Tenenbaum Pearson Education Inc, 2nd edition, 2015.

Reference Book:

1:“Data Structures and Program Design in C”, Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019.

UE21CS241A: STATISTICS FOR DATA SCIENCE (4–0–0–4)

No. of Hours: 75

Class #	Chapter Title/Reference Literature	Topics to be covered	% of Portion	
			% of syllabus	Cumulative
1.	**Unit: 1** **Introduction to Data Science, Statistics and Visualizing data** **T1: Chapter 1 1.1-1.3**	**Introduction to Data Science:** Motivating Examples and Scope.	28%	28%
2.		**Sampling:** Introduction, Sample, Population, Types of population – Tangible, Conceptual. (1.1),		
3.		Sampling Methods (1.1)		
4.		Sampling Methods (1.1) Sampling Errors – Handout 1		
5.		Types of Data, Types of Experiments – Controlled and Observational study (1.1)		
6.		**Hands on session: Purpose and Components of Python-A quick recap**		
7.		**Getting and Analyzing Data:**Scraping the Web (Handout 2) **Hands on Session-Web Scraping with BeautifulSoup**		
8.		**Getting and Analyzing Data:** Reading Files (.csv) (Handout 3) **Hands on Session Reading Files with Pandas**		
9.		**Data Cleaning:** Need for Data Cleaning, Basics of Data Cleaning.(Handout 4) **Hands on Session- Data Cleaning with Pandas**		
10.		**Statistics :**Introduction, Types of Statistics, Summary Statistics(1.2)		
11.		Summary Statistics (cont.), Statistic and Parameter.(1.2)		
12.		**Hands on Session-Summary Statistics and Sampling Methods**		
13.		**Data Visualization and Interpretation :** **Graphical summaries -** Histogram – Equal and Unequal Widths (1.3) **Hands on Session-Data Visualization with Matplotlib and Seaborn**		
14.		Visualizing Data: Box Plots (1.3) **Hands on Session-Data Visualization with Matplotlib and Seaborn**		
15.		Visualizing Data: Two variables (Scatter Plots) (1.3), **Hands on Session-Data Visualization with Matplotlib and Seaborn**		
16.		Visualizing Data: Bar Charts – Handout 5, **Hands on Session-Data Visualization with Matplotlib and Seaborn**		

17.		Heat Maps-Handout 6 **Hands on Session-Data Visualization with Matplotlib and Seaborn** Good vs. Bad Visualization.(Handout 7)		
18.		**Tutorial 1: Sample Exercise Problem Solving**		
19.		**Revision/ISA 1**		
20.		**Case Study 1: Real world Application**		
21.		**Coding Assignment 1 based on Case Study 2**		
22.	**Unit: 2** **Random Variables and Probability Distributions** **T1: Chapter 2 2.4 – 2.5, Chapter 4 4.1 – 4.3, 4.5**	Brief overview of Probability Basics.(Handout 8) **Random Variables** : Introduction, Discrete Random Variables(2.4)	20%	48%
23.		Continuous Random Variables(2.4)		
24.		Chebyshev's inequality(2.4),		
25.		Linear Functions of Random Variables.(2.5)		
26.		Linear Functions of Random Variables.(2.5)		
27.		**Probability Distributions:** TheBernoulli Distribution(4.1)+derivation of Mean and Variance		
28.		The Binomial Distribution(4.2) **Hands on session on distributions**)+derivation of Mean and Variance		
29.		The Poisson Distribution(4.3) **Hands on session on distributions**		
30.		The Normal Distribution(4.5)		
31.		The Normal Distribution(4.5) **Hands on session on distributions**		
32.		Generation of Random Variates (Handout 9) **Hands on session**		
33.		**Tutorial 2: Sample Exercise Problem Solving**		
34.		**Revision/ISA 2**		
35.		**Case Study 1: Real world Application**		
36.		**Coding Assignment 2 based on Case Study 2**		
37.	**Unit: 3** **Probability Distributions and Confidence Intervals**	Principles of Point Estimation : Mean squared error(4.9)	20	68
38.		Maximum likelihood estimate (4.9)+(Handout 10)		
39.		Maximum likelihood estimate (4.9)+(Handout 10) contd.,		
40.		Normal Probability Plot (4.10) **Hands on session**		
41.		Sampling Distribution		
42.		Sampling concepts : The Central Limit Theorem and its applications(4.11) **Hands on session**		
43.		Continuity Correction-Normal approximation to Binomial and Poisson distribution **Hands on session**		
44.		**Confidence Intervals:**Introduction, Interval		

	T1: Chapter 4 4.9 – 4.11 Chapter 5 5.1-5.4,5.7	estimates for proportion of large samples. (5.2)		
45.		Confidence intervals for mean of Small Samples.(5.3) Student's t Distribution **Hands on session**		
46.		Confidence Intervals for the Difference Between Two Means for large samples(5.4),		
47.		Confidence Interval estimates for paired data.(5.7) Factors affecting Margin of Error.(Handout 11) **Hands on session**		
48.		**Tutorial 3: Sample Exercise Problem Solving**		
49.		**Revision/ISA 3**		
50.		**Case Study 1: Real world Application**		
51.		**Coding Assignment 3 based on Case Study 2**		
52.	**Unit: 4 Hypothesis and Inference. T1: Chapter 6 6.1 – 6.3, 6.5, 6.9, 6.10, 6.12**	**Hypothesis Testing:** Introduction (6.1)	17.33	85.33
53.		Large sample tests for a Population Mean (6.1)		
54.		Drawing conclusions from the results of Hypothesis tests(6.2)		
55.		Large sample tests for a Population proportion (6.3)		
56.		Large - Sample tests for Difference between two means (6.5)		
57.		**Hands on session :Hypothesis Testing**		
58.		Distribution Free Tests. (6.9)		
59.		Chi-squared Test.(6.10)		
60.		Fixed Level Testing (6.12), Type I and Type II Errors (6.12)		
61.		**Tutorial 4: Sample Exercise Problem Solving**		
62.		**Revision/ISA 4**		
63.		**Case Study 1: Real world Application**		
64.		**Coding Assignment 4 based on Case Study 2**		
65.	**Unit: 5 Power of Test T1: Chapter 6 6.13 Simple Linear Regression.**	Power of a Test.(6.13)	14.67	100
66.		Power of a Test.(6.13)		
67.		Factors affecting Power of a Test.(Handout 12)		
68.		**Simple Linear Regression:** Introduction**,** Correlation.(7.1)		
69.		The Least squares Line.(7.2)		
70.		Predictions using regression models - Uncertainties in Regression Coefficients.(7.3)		
71.		Checking Assumptions and transforming data.(7.4) **Hands on session on Linear regression**		
72.		**Tutorial 5: Sample Exercise Problem Solving**		
73.	**T1: Chapter 7**	**Revision/ISA 5**		

74.	***7.1 – 7.4***	**Case Study 1: Real world Application**		
75.		**Coding Assignment 5 based on Case Study 2**		

Literature:

Book Type	Code	Title & Author	Publication Information		
			Edition	**Publisher**	**Year**
Text Book	T1	Statistics for Engineers and Scientists, William Navidi.	4th	McGraw Hill Education, India	2013
Reference Book	T2	The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Raj Jain		Wiley	2008
Reference Book	T3	Data Science From Scratch, Joel Grus	1st	O'Reilly	2015
Reference Book	T4	Sampling- Design and Analysis, Sharon L. Lohr	2nd	Cengage	2010
Reference Book	T5	Statistics for Engineers and Scientists, William Navidi.	3rd	McGraw Hill Education, India	2010

UE21CS242A : WEB TECHNOLOGIES (4–0–0–4)

No. of Hours: 75

Class #	Chapter Title/Reference Literature	Topics to be Covered	% of Portions Covered	
			% of Syllabus	Cumulative %
1.	**UNIT 1(12 Hours) TB1 : Chapter 1,Chapter 13 : Pageno 309 to 323 Chapter 18 and 19**	Introduction to Web Architecture and Web protocols (HTTP Request Response Formats, URLs)	**21.43%**	**21.43%**
2.		Basic Mark-ups & syntax, HTML elements & attributes		
3.		Web Form 2.0 & Form Controls		
4.		HTML5 (New Tags, Inputs, Elements and Controls),		
5.		CSS3.0-Styles and Style sheets,		
6.		Selectors, Style properties		
7.		Box Model and Positioning		
8.		CSS Flex Property and Media Queries		
9.		JavaScript Basics(Variables, Scope)		
10.		JavaScript Basics: Functions, Hoisting		
11.		JavaScript Built in Objects		
12.		JavaScript Objects		
13.		Review: Lab Assignment on HTML and CSS: Unit 1 – Submission of project Titles and Team Members		
14.		ISA 1		
15.	**UNIT 2(12 hours) TB1 : Chapter 17, Chapter 21,22,24,25**	JavaScript Objects and Prototypal Inheritance	**21.43%**	**42.86%**
16.		DOM Manipulations		
17.		Events		
18.		Event Handling in JavaScript,		
19.		HTML5 (APIs), Audio, Video, Progress		
20.		HTML5 (APIs) – Canvas, SVG, File api, geolocation, web workers.		
21.		JQuery (Introduction, Handling events)		
22.		JQuery (Introduction, Handling events)		
23.		Callbacks & Promises, Single Page Application		
24.		XML Vs JSON, Asynchronous Communication- XHR (properties and methods)		
25.		$.ajax,$.get,$.post,		
26.		$load, Fetch API		
27.		Review: Lab Assignment on Javascript : Unit 2		
28.		ISA 2		
29.		Project Review- Submission of wireframe design		
30.	**UNIT 3(10 Hours) TB2 : Chapter 1,3,4**	MERN Introduction	17.85%	**60.71%**
31.		React Classes and Components		

32.	,8	Styling and Complex components		
33.		Properties, States and Context		
34.		Component lifecycle methods		
35.		Self Learning: Explore on Bootstrap template. Apply Bootstrap's choices of color, size, font and layout to your own project.		
36.		Stateless components		
37.		Refs & Keys		
38.		Event Handling		
39.		React Form Handling		
40.		React Form Handling		
41.		Review: Lab Assignment on ReactJS: Unit 3		
42.		ISA 3		
43.	**UNIT 4(12 Hours) TB2 : Chapter 6**	Understanding Node JS Architecture	**21.43%**	**82.14%**
44.		Set up Node JS app		
45.		Node Modules		
46.		Buffers, Streams, File system		
47.		HTTP Module, Handling HTTP Requests		
48.		HTTP Module, Handling HTTP Requests		
49.		Self Learning : Compare AngularJS and ReactJS and NodeJS		
50.		Mongo DB- Documents, Collections		
51.		Reading and Writing to DB		
52.		MongoDB Node JS Driver		
53.		Events and Events Emitter		
54.		Running a react application on NodeJS(Hands-on)		
55.		React Router		
56.		Review: Lab Assignment on NodeJS: Unit 4		
57.		ISA 4		
58.	**UNIT 5(10 hours)TB2 : Chapter 5**	Introduction to Web services	**17.86%**	**100%**
59.		REST API's		
60.		Express Framework Overview		
61.		Routing		
62.		URL Binding		
63.		Error Handling		
64.		Express Middleware		
65.		Form Data		
66.		File Upload		
67.		File Upload		
68.		Review: Lab Assignment on ExpressJS: Unit 5		
69.		Self Learning: Vue JS		
70.		ISA 5		
71.		Guest Lecture		
72.				
73.		Mini Project Presentation and Evaluation		
74.				
75.				

Tools / Languages:

HTML, CSS, JavaScript, MERN Technologies.

Books:

Book Type	Code	Title & Author	Publication Information		
			Edition	Publisher	Year
Text Book	T1	Learning PHP, MySQL & JavaScript, 5th Edition. by Robin Nixon. May 2018, O'Reilly Media, Inc. ISBN: 9781491978917	2nd	Wiley Publishing	2018
Text Book	T2	Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node by Vasan Subramanian. March 2017, Apress	1st	O'Reilly	2017
Reference Book	R1	Beginning Node.js, Express & MongoDB Development by Greg Lim, July 2019	1st	McGraw Hill	2019
Reference Book	R2	Learning React, Functional Web Development with React and Redux By Alex Banks and Eve Porcello, May 2017, O'Reilly Media	1st	O'Reilly Media	2017

UE21CS243A : Automata Formal Languages and Logic (4–0–0–4)

No. of Hours: 75

Hours	Topic	Chapter and Section	Coverage	
			% of Syllabus	Cumulative %
Unit 1: Introduction				
1.	Mathematical Preliminaries and Notation,	T1 – 1.1 T2 – 1.2	18	18
2.	Three Basic Concepts. Finite Automata: Deterministic Finite Accepters	T1 – 2.1		
3.	Deterministic Finite Accepters	T1 – 2.1		
4.	Deterministic Finite Accepters	T1 – 2.1		
5.	Non-Deterministic Finite Accepters,	T1 – 2.2		
6.	Non-Deterministic Finite Accepters,	T1 – 2.2		
7.	Equivalence of Deterministic and Non-Deterministic Finite Accepters,	T1 – 2.3		
8.	Equivalence of Deterministic and Non- Deterministic Finite Accepters,	T1 – 2.3		
9.	Reduction of the number of states in Finite Automata.	T1 – 2.4		
10.	Reduction of the number of states in Finite Automata.	T1 – 2.4		
11.	Unit wise revision session			
12.				
13.	**ISA 1**			
Unit 2: Regular Languages and Grammars				
14.	Regular Expressions.	T1 – 3.1	18	36
15.	Regular Expressions	T1 – 3.1		
16.	Connection between Regular Expressions and Regular Languages.	T1 – 3.2		
17.	Equivalence of two Regular Expressions,	T1 – 3.2		
18.	Regular Grammars	T1 – 3.3		
19.	Regular Grammars, Equivalence of Regular Grammar and Finite Automata	T1 – 3.3		
20.	Equivalence of Regular Grammar and Finite Automata	T1 – 3.3		
21.	Properties of Regular Languages Elementary Questions about RegularLanguages	T1 – 4.1 T1 – 4.2		
22.	Pumping Lemma	T1 – 4.3		
23.	Pumping Lemma, Identifying NonRegular Languages.			
24.	Unit wise revision/case study session			
25.				

26.	**ISA 2**			
27.	Definitions of PDA and CFL, Deterministic Pushdown Automata,	T1 – 7.1	21.3	57.3
28.	Deterministic Pushdown Automata,	T1 – 7.1		
29.	Non-Deterministic Pushdown Automata,	T1 – 7.2		
30.	Non-Deterministic Pushdown Automata,	T1 – 7.2		
31.	Pushdown Automata and Context Free Languages,	T1 – 7.2		
32.	Context Free Grammars,	T1 – 5.1		
33.	Context Free Grammars,	T1 – 5.1		
34.	Parsing and Ambiguity. Simplification of Context–Free Grammars.	T1 – 5.2		
35.	Normal Forms: Methods for Transforming Grammars, TwoImportant Normal Forms: Chomsky Normal Form	T1 – 6.1 T1 – 6.2		
36.	CYK Algorithm	T1 – 6.3		
37.	Greibach Normal Form	T1 – 6.2		
38.	CFG to PDA	T1- 7.2		
39.	Unit wise revision session			
40.				
41.	**ISA 3**			
Unit 4: Properties of Context-Free Languages				
42.	Properties of Context-Free Languages:Closure Properties and Questions about Context–Free Languages,	T1- 8.2	21.3	78.6
43.	Pumping Lemma for Context–Free Languages.	T1- 8.1		
44.	Pumping Lemma for Context–Free Languages.	T1- 8.1		
45.	Turing Machines: The Standard TuringMachine, Constructing Turing Machines,	T1- 9.1		
46.	Constructing Turing Machines,	T1- 9.1		
47.	Combining Turing Machines for Complicated Tasks,	T1- 9.2		
48.	Combining Turing Machines for Complicated Tasks, Turing's Thesis.	T1- 9.2 T1- 9.3		
49.	Hierarchy of Formal Languages and Automata: Recursive and Recursively	T1- 11.1		
	Enumerable Languages, the Chomsky Hierarchy.			
50.	Limits of Algorithmic Computation: Some Problems that cannot be solved by Turing Machines	T1- 12.1 T1- 12.3		
51.	Undecidable Problem for Recursively Enumerable Languages,	T1- 12.2		
52.	Undecidable Problem for Recursively	T1- 12.2		

	Enumerable Languages, idea ofreduction	T1- 11.3		
53.	Unit wise revision session			
54.				
55.	**ISA 4**			
Unit 5: Propositional Logic				
56.	A very simple Logic, Syntax, Semantics,	T2- 7.4, T2-7.4.1, T2-7.4.2	21.3	100
57.	A simple knowledge Base, A simple inference procedure.	T2- 7.4.3, T2-7.4.4		
58.	Propositional Theorem Proving: Inference and Proofs,	T2- 7.5.1, 7.5.2		
59.	Proof by Resolution	T2- 7.5.1, 7.5.2		
60.	Conjunctive Normal Form	T2- 7.5.1,		
61.	Conjunctive Normal Form	T2- 7.5.1		
62.	A resolution algorithm.	T2- 7.5.2		
63.	Syntax and Semantics of First Order Logic:	T2- 8.2		
64.	Models for First Order Logic Symbols and interpretations,	T2- 8.2		
65.	Terms, Atomic Sentences, Complex Sentences	T2- 8.2		
66.	Quantifiers, Equality, Numbers	T2- 8.3.3		
67.	Sets and Lists. Example - The electronic circuits' domain.	T2- 8.4.2		
68.	Problems Based on CNF and Resolutionalgorithm			
69.	**ISA 5**			
70.	Unit wise revision session			
71.	Introduction to Prolog Programming			
72.	**Assignment 2: Syntax Validation of a programming language by writing theContext Free Grammar. (PLY, ANTLR Tools)**			
73.				
74.	**Assignment 1: Implementation of RegEx for NLP Applications**			
75.				

Tools:

JFLAP - Java Formal Languages and Automata Package

Text Book(s):

1. “An Introduction to Formal Languages and Automata”, Peter Linz, Jones and Bartlett,New Delhi, India, 5th Edition, 2011.

2. Artificial Intelligence – A Modern Approach”, Stuart Russell and Peter Norvig,Pearson,3rd Edition (Paperback),2016

References:

1. “Theory of Computation”, Michael Sipser, Cengage Learning, New Delhi, India, 2008.

2. “Introduction to Automata Theory, Languages, and Computation”, John E Hopcroft, Rajeev Motwani, Jeffrey D Ullman, Pearson Education, New Delhi, India, 3rd Edition, 2009.

3. “Theory of Computation: A Problem–Solving Approach”, Kavi Mahesh, Wiley India, NewDelhi, 2012