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.

IV SEMESTER (2021-25 BATCH)

Sl. No.	Course Code	Course Title	Hours per week				Credits	Tools / Languages	Course Type
			L	T	P	S	C		
1	UE21CS251B	Microprocessor and Computer Architecture%	4	0	2	5	5	ARM Simulator, Aurdino Microprocessor Kit, Paracache Simulator.	CC-Lab Integrated
2	UE21CS252B	Computer Networks	4	0	2	5	5	Wireshark, Python	CC-Lab Integrated
3	UE21CS241B	Design and Analysis of Algorithms	4	0	1	4	4	C-Programming Language, GCC Compiler	CC-Independent
4	UE21CS242B	Operating Systems@	4	0	1	4	4	Pthread, C Programming Language, Linux/ Unix OS for system call implementation.	CC-Independent
5	UE21MA241B	Linear Algebra	4	0	1	4	4		CC-Independent
6	UE21MA221B*	Bridge Course Mathematics –II (Applicable to Lateral Entry Students)	2	0	0	2	0		FC-Independent
Total			20/22	0	7	22/ 24	22		

Note: Desirable Knowledge - %UE21CS251A, @UE21CS252A.
* - Audit Course

UE21CS251B - Microprocessor and Computer Architecture (4-0-2-5-5)

No. of Credits: 5 **# of Hours: 105**

Class #	Chapter Title / Reference Literature	Topics to be Covered	% of portions covered	
			Reference Chapter	Cumulative
		UNIT 1: Introduction to Microprocessor Architecture &ISA		
1	1.6,2.3 of T2, A-3 of T1, pg no: 51-55 of T2 Chapter 3.1 to 3.5 of T3 6.8,5.6 of T2	Introduction and Motivation. How Program Execute? Relation between Processor, Operating System, Compiler and Memory. 1.1 Interrupts, Context Switching an overview. 1.2 Classification CISC Vs RISC and Introduction to ARM Processor	22%	22%
2		ARM Processor: Register set, Introduction to ARM ISA and Instruction Layout		
3		Data Processing Instructions: Addition and Subtraction with programming Examples		
4		Data Processing Instruction variants		
5		Data Transfer Instructions: Load and Store with programming examples		
6		Data Transfer instruction and STACK operations		
7		Branch Instructions		
8		Multiplication Instructions and Instruction Encoding		
9-10 L1		Implementation of ARM7TDMI-ISA to Block transfer of data items, Find sum of N data items in the memory.		
11-12 L2		Find the product of two 32bit numbers using barrel shifter		
13		Interrupts and Programming Examples		
14-15 L3		Convert the statement in C language into an ALP. Find Factorial, GCD, search for an element, sum of n elements in an array using various addressing modes		
16 A1.1		Write a program in ARM7TDMI-ISA to search for an element in an array. Display appropriate messages on the standard output device.For Successful search display as “Successful Search” and if the search is unsuccessful, display as “Unsuccessful Search”. Use Binary search Technique.		
17 A1.2		Write a program in ARM7TDMI-ISA to find a sub string in a given main string. Example1: Main string : My name is Bond. Character : ‘name’. Expected Output : “String Present” Example2: Main string : My name is Bond. Character : ‘James’. Expected Output : “String Absent”		
18		Instruction Encoding 1: Data Processing Instruction		

19		Instruction Encoding 2: Data Transfer Instruction		
20		Instruction Encoding 3: Branch and other Instructions		
21		Revision		
UNIT 2 Pipelining				
22	4.1,4.2 of Text T2 Appendix C-1, C-2, Sec 1.1 , 1.4, 1.5 of T1	Introduction to Pipelining (3 & 5 Stage)	20%	42%
23		Performance Analysis, Speed up Calculations…..etc		
24		Introduction to Pipeline hazards, Structural Hazards		
25 A2.1		Consider the following sequence of instructions in MIPS architecture. LDR R1, [R2,#40] ADD R2, R3, R3 ADD R1, R1, R2 STR R1, [R2,#20] a. Find all dependencies in this instruction sequence. b. Find all hazards in this instruction sequence for a five stage pipeline with and without data forwarding. c. Find whether NOPs are required to be introduced inspite of data forwarding in this instruction sequence.		
26-27 L4		Implementation of ARM7TDMI code to generate Fibonacci series, smallest, largest in an array.		
28		Data Hazards 1		
29		Data Hazards 2		
30		Control Hazard 1		
31-32		Control Hazard 2 & Introduction to Branch Prediction		
33 A2.2		Consider the following sequence of instructions in MIPS architecture. LDR R1, [R6,#40] BEQ R2, R3, LABEL2 ; BRANCH TAKEN ADD R1, R6, R4 LABEL2:BEQ R1,R2, LABEL1 ; BRANCH NOT TAKEN STR R2,[R4, #20] AND R1, R1, R4 a. Draw the pipeline execution diagram for this code, assuming there are no delay slots and that branches execute in the EX stage. b. Repeat the exercise mentioned in a and draw the pipeline execution diagram for this code, assuming that delay slots are used by writing a "SAFE INSTRUCTION" in the delay slot.		
34-35 L5		Usage of Multiple Load and Store instructions to perform the parameter passing techniques to a function.		
36		Branch Prediction 1		
37		Branch Prediction 2		
38		Branch Prediction 3		
39		Performance Analysis, Speed up calculations …..etc		
40-41 L6		Implementation of Matrix Operations – Addition, Multiplication.		

42		Revision		
		UNIT 3:Memory Hierarchy		
43	Appendix B.1, B.2, B.3 of T1	Introduction to Memory Subsystem, Bottle neck, Memory Hierarchy Introduction to Cache, Locality of reference and Cache Design Philosophy	20%	62%
44		Cache Design Philosophy Continued: Block Placement, Block Identification, BlockReplacement, Read / Write issues with cache		
45		Direct Map Cache Memory		
46-47 L7		Demonstration of MIPS5 simulator to understand pipeline architecture		
48		Set Associative Cache Memory		
49		Fully Associative Cache Memory		
50 A3.1		How many total bits are required for a direct –mapped cache with 16KB of data and 4 word blocks, assuming a 32-bit address?		
51		Page Replacement Algorithms		
52		Read / Write Policy		
53-54 L8		Working with Memory simulator : PARACACHE.		
55		Performance Analysis		
56		1st Optimization		
57		1st Optimization&2nd Optimization with examples		
58		3rd Optimization with examples		
59		4th Optimization with examples		
60-61 L9		Demonstration of Direct mapping cache and Associative cache.		
62 A3.2		Consider a cache with 64 blocks and a block size of 16bytes. To what block number does the byte address 1200map?Assume all are decimal numbers.		
63		Revision		
		UNIT 4 :Memory Optimization Continued &IO Sub system		
64	Appendix B.3 of T1	5th Optimization with examples		
65		5th Optimization with examples		
66		6th Optimization with examples		
67		6th Optimization with examples		
68 A4.1		Increasing associativity requires more comparators and more tag bits per cache block. Assuming ac cache of 4K blocks, 4 word block size, and a 32-bit address, find the total number of sets and the total number of tag bits for caches that are direct mapped, two-way and four-way set associative , and fully associative.		
69-70 L10		Working with Memory simulator : PARACACHE and demonstration of Set Associative memory write with all options.		
71-72		Memory Introduction to flash storage, Connecting Processors, and I/O devices.		

<table>
<tr><td>73</td><td rowspan="4"></td><td>Interfacing I/O Devices to the Processor</td><td rowspan="8">16%</td><td rowspan="8">78%</td></tr>
<tr><td>74</td><td>DMA Controller</td></tr>
<tr><td>75</td><td>Memory and Operating System</td></tr>
<tr><td>76</td><td>Examples</td></tr>
<tr><td>77
A4.2</td><td></td><td>Recall that we have two write policies and write allocate policies, their combinations can be implemented in either in L1 or L2 cache.
<table>
<tr><td></td><td><b>L1 cache:</b></td><td><b>L2 cache:</b></td></tr>
<tr><td>a</td><td>Write back, write allocate</td><td>Write –through, non wr allocate</td></tr>
<tr><td>b</td><td>Write back, write no allocate.</td><td>Write –through, write allocate</td></tr>
</table>
i. Describe the procedure of handling an L1 write miss, considering the component involved and the possibility of replacing a dirty block.
ii. For a multilevel exclusive cache(a block can only reside in one of the L1 and L2 caches) configuration, describe the procedure of handling an L1 write miss, considering the component involved and the possibility of replacing a dirty block.</td></tr>
<tr><td>78-79
L11</td><td></td><td>Working with Memory simulator : PARACACHE Demonstration Fully Associative mapping, memory write with all options.</td></tr>
<tr><td>80-81
L12-P</td><td></td><td>Introduction to Arduino Board: Working of various Sensors with Arduino board.</td></tr>
<tr><td>80</td><td></td><td>Revision</td></tr>
<tr><td colspan="5"><b>UNIT 5: Advances in Architecture</b></td></tr>
<tr><td>81</td><td rowspan="11">Sec 1.9 of T1, Sec 3.1, 4.1</td><td>Introduction to Parallel Computing</td><td rowspan="12">22%</td><td rowspan="12">100%</td></tr>
<tr><td>82</td><td>Parallel Computing : Introductory concepts and terminology-Flynn’s taxonomy,</td></tr>
<tr><td>83</td><td>Parallel computing memory architectures,</td></tr>
<tr><td>84</td><td>parallel programming models</td></tr>
<tr><td>85</td><td>parallel examples: matrix multiplication</td></tr>
<tr><td>86</td><td>Amdahl’s Law, Gustafson Law,</td></tr>
<tr><td>87</td><td>Hardware Multi threading</td></tr>
<tr><td>88</td><td>Multi-Core Architecture</td></tr>
<tr><td>89</td><td>Multi-Core Architecture continued..</td></tr>
<tr><td>90</td><td>Introduction to GPU Computing</td></tr>
<tr><td>91-100
L13-P</td><td>Project Work using sensors</td></tr>
<tr><td>101-
105</td><td></td><td>ISA1-ISA5(CBT)</td></tr>
</table>

Literature:

Book Type	Code	Title & Author	Publication Info		
			Edition	Publisher	Year
Text Book	T1	Hennessy Patterson	Fifth Edition	MK Morgan Kaufmann	2012
Text Book	T2	ARM System on Chip, Steve Furber	Second Edition,	Pearson Education	2000
Text Book	T3	ARM System Developer's Guide	Reprint 2009	Elsevier	2009

UE21CS252B: Computer Networks (4-0-2-5-5)

No. of Credits: 5 **# of Hours: 105**

Class #	Chapter Title /Reference Literature	Topics to be Covered	% of Portion covered	
			% of Syllabus	Cumulative %
Unit – 1 Computer Networks and the Internet				
1	1.1.1	Introduction to computer networks, What is internet? A Nuts-and-Bolts description	16%	16%
2	1.1.2, 1.1.3	A services description, What is a Protocol?		
3	1.2.1	Network edge: Access networks		
4	1.2.2	Physical media		
5				
6	Lab-1	[Basic Commands]		
7				
8	1.3.1	Network core: Packet switching		
9	1.3.2, 1.3.3	Circuit switching, Network of networks		
10	1.4.1	Overview of delay in Packet-switched networks		
11	1.4.2	Queuing delay and Packet loss		
12	1.4.3, 1.4.4	End-to-End delay, Throughput in computer networks		
13	Practice Session	Numerical Problems [Individual] Work By Hand		
14				
15	1.5 (T1) 2.1, 2.2 (R1)	Protocol layers - The OSI model		
16	2.3 (R1) 1 (R2)	TCP/IP protocol suite		
Unit – 2 Application Layer				
17	2.1.1, 2.1.2	Network application principles: Network application architectures, Processes communication	20%	36%
18	2.1.3	Transport services available to applications		
19	2.1.4	Transport services by Internet		
20	Lab-2	[Cisco Packet Tracer - Topology Creation]		
21				
22	ISA – 1			
23	2.2.1, 2.2.2	The web and HTTP, Non-persistent and Persistent connection		
24	2.2.3	HTTP message format		
25		HTTP vs HTTPS		
26	2.2.4	Cookies		
27	2.2.5	Web Caching		
28	Lab-3	[HTTP Persistant Connection, Non-Persistant Connection, Cookies, Web Server Setup]		
29				
30	2.4.1, 2.4.2, 2.4.3	DNS – Services provided, Overview of how DNS works, DNS records and messages		
31	2.5.1	Peer–to–Peer applications		
32	2.7.1	Socket Programming with UDP		
33	2.7.2	Socket Programming with TCP		

34	20, 21 (R1)	Other Application Layer Protocols: FTP, SMTP, SNMP		
35	23, 24 (R1)	Telnet, SSH		
36	Lab-4	DNS		
37				
38	ISA – 2			
Unit – 3 Transport Layer				
39	3.1	Introduction to transport layer, Relationship between transport and network layer, Overview of the transport layer in the Internet	16%	52%
40	3.2	Multiplexing and Demultiplexing		
41	3.3	Connectionless transport: UDP, Segment structure, Checksum		
42	3.4.1	Principles of reliable data transfer, Building a reliable data transfer protocol		
43	3.4.2	Principles of Reliable Data Transfer protocol – RDT 2.0		
44				
45	3.4.3	Principles of Reliable Data Transfer protocol – RDT 3.0		
46				
47	3.4.4	Pipelined Protocols: Go–Back–N, Selective Repeat		
48	3.5.1, 3.5.2, 3.5.3, 3.5.4	Connection Oriented Transport: TCP, The TCP connection, TCP segment structure		
49				
50	3.5.5, 3.5.6	Connection oriented transport TCP reliable data transfer, TCP Flow control & TCP connection management (hands on)		
51				
52	3.6	Principles of congestion control, TCP congestion control		
53	ISA – 3			
UNIT – 4 NETWORK LAYER				
54	Assignment-1I	Socket Programming [TCP/UDP] (Chat/File Transfer/Time Server etc.,) [In Group of 2/3] - 6 Hrs - 8 Marks	22%	74%
55				
56	T1: 4.1	Overview of network layer, Forwarding and routing, Network service models		
57	4.2.1, 4.2.2	Inside router: Input port processing and Destination-based forwarding, Switching		
58	4.2.3, 4.2.4	Output port processing, where does Queuing occur?		
59	4.2.5	Packet scheduling		
60	4.3.1	The Internet Protocol – IPv4, Datagram format (hands on)		
61	4.3.2	Fragmentation		
62	Assignment-1 II Continuation	Socket Programming [TCP/UDP] (Chat/File Transfer/Time Server etc.,) [In Group of 2/3] - 6 Hrs - 8 Marks		
63				
64	4.3.3	IPv4 Addressing		
65	4.3.3	IPv4 Addressing		
66	4.3.4	IPv4 Addressing, NAT		
67	26.1, 26.2,	IPv6 Addressing: Introduction, Address space allocation,		

	26.3, 27.1 (R1)	Packet format (hands on)		
68	27.2, 27.3 (R1)	Transition from IPv4 to Ipv6, Tunnelling		
69	4.3.3	Network layer protocols: DHCP (hands on)		
70		ICMP (hands on)		
71	5.2.1	Introduction to routing algorithms: Link state		
72	5.2.2	Distance vector		
73	5.2.2	Problems		
74	ISA – 4			
75	Assignment-1 III Continuation	Socket Programming [TCP/UDP] (Chat/File Transfer/Time Server etc.,) [In Group of 2/3] - 6 Hrs - 8 Marks		
76				
UNIT – 5 LINK LAYER AND LAN				
77	T1: 6.1, 6.2: 6.2.1	Introduction to link layer, Error-detection and correction techniques: Parity checks, Internet checksum, Cyclic redundancy check	26%	100%
78	6.2.2	Multiple access protocols: CSMA/CD		
79	6.2.3	Switched LAN: Link layer addressing & ARP		
80	6.4.1	Ethernet		
81	Industry Problem-1 I	Development of Port Scanner, Web Scanner, OS Finger Printing etc- 6Hrs – 8 Marks [In Group of 2/3]		
82				
83	6.4.2	Link-layer switches		
84	6.4.3	Retrospective: A day in the life of a web page request		
85	6.4.4	Physical layer: Purpose, Signals to Packets		
86	6.7	Analogvs Digital Signals, Transmission media		
87	Lab-5	IPv4 Addressing &IPv6 Addressing		
88				
89	7.3 (T1) 3.2 (R1)	Wireless LANs: IEEE 802.11 LAN architecture		
90				
91	7.3.2	802.11 MAC protocol		
92	7.3.3	IEEE 802.11 Frame		
93	Industry Problem-1 II	Development of Port Scanner, Web Scanner, OS Finger Printing etc		
94				
95	Lab-6	ICMP Redirect, TTL Expiry, PMTU Discovery		
96				
97	Industry Problem-1 III	Development of Port Scanner, Web Scanner, OS Finger Printing etc		
98				
99	Lab Quiz			
100	Industry Problem-2 (Extra Hrs)	RFC Reading Exercise [Each student to get a different protocol within a section] 2 Marks		
101				
102	Industry Problem Evaluation			
103				
104				
105	ISA – 5			

Lab / Hands-on: 18 Hours

1. Program on ping, tcpdump and wireshark.
2. Program on Exploring HTTP with wireshark, Web Server setup, FTP/SMTP and SNMP Clients, Telnet,
SSH and DNS
3. Program on Wireshark based TCP congestion window plotting, UDP traffic analysis.
4. Program on Cisco Packet Tracer based Router experiments; IPv4 Fragmentation based wireshark
experiments, Inspection of DHCP, ICMP.
5. Program on IPv6 Packets using wireshark.
6. Program on Wireshark based Link Layer protocol inspection.

Tools/ Languages: Wireshark, Python.

Literature:

Book Type	Code	Title & Author	Publication Info		
			Edition	Publisher	Year
Text Book		“Computer Networking: A Top-Down Approach”, James F. Kurose, Keith W. Ross	7th Edition	Pearson Publication	2017
Reference Book		“TCP IP Protocol Suite”, BehrouzForouzan,	4th Edition	McGraw-Hill,	2010

UE21CS241B: Design and Analysis of Algorithms (4-0-1-4-4)

No. of Credits: 5 **# of Hours: 75**

Class #	Chapter Title / ReferenceLiterature	Topicsto beCovered	% of portion covered	
			% ofSyllabus	Cumulative%
1	**Unit#1 Introduction**	Motivation for the course.	**22**	**22**
	T1: Chapters1.1,1.2,1.3, 2.1,2.2,2.3,2.4	Evaluation policy of the course. Introduction to Algorithms.		
2		Fundamentals of Algorithmic problem-solving. Important problem types–sorting, searching. **Practice:** Matrix Addition, Matrix Multiplication		
3		Important problem types–string processing,		
		Graph problems, Combinatorial ,Geometric,		
		Numerical problems. **Practice:** Use header files, modular programming		
4		Analysis Framework Orders of Growth **Practice:** Function calls, pointers to functions etc.		
5		**Ticking Session**		
6		Asymptotic Notations Basic Efficiency Classes		
7		Using Limits for comparing order of growth Mathematical Analysis of Non-recursive Algorithms **Practice:** Element Uniqueness problem		
8		Mathematical Analysis of Non-recursive Algorithms **Practice:** Naïve implementation of Travelling Salesman Problem		
9		Solving Recurrences of Recursive Algorithms		
10		**Ticking Session**		
11		Mathematical Analysis of Non-recursive Algorithms		
12		Performance Analysis Vs Performance Measurement **Practice:** Tower of Hanoi		
13		**Practice:** n^{th} Fibonacci number Generation – using second-order recurrence with constant coefficients.		

14		**Ticking Session**		
15		**ISA 1**		
16	**Unit#2** **Brute Force and Divide-and-Conquer**	Brute Force: Selection Sort, Bubble sort **Practice:** Bubble Sort Implementation	23	45
17		Sequential Search Brute-Force String Matching **Practice:** Selection sort implementation		
18		Knapsack Problem ,Assignment Problem		
19		Exhaustive Search Travelling Salesman Problem **Practice** Naïve String Match implementation		
20		**Ticking Session**		
21		Divide and Conquer Approach, General Divide and Conquer Recurrence, Master Theorem		
22		Solving Recurrences using Master Theorem		
23		Merge Sort **Practice:** Merge Sort Implementation		
24		Quick Sort **Practice:** Quick Sort Implementation		
25		**Ticking Session**		
26		Binary Search		
27		Binary Tree Traversals Complexity analysis for finding the height of BST		
28		Multiplication of Large Integers		
29		Strassen’s Matrix Multiplication		
30		**ISA 2**		
31	**Unit #3:** **T1:Chapters 5.1,5.3,5.4 6.1,6.3,6.4, 7.4**	Decrease-and-Conquer approach–Insertion Sort	20	65
32		Depth First Search Topological Sorting **Practice:** DFS implementation		
33		Algorithms for Generating combinatorial Objects: Generating Permutations Johnson Trotter Algorithm **Practice:** Johnson Trotter Algorithm implementation		
34		Generating Subsets		
35		**Ticking Session**		
36		Decrease-by-a-Constant-Factor Algorithms: Fake coin Problem, Russian Peasant Method for Multiplication ,Josephus Problem		

37		Transform-and-Conquer Approach Pre-sorting **Practice:** Topological Sort implementation		
38		Heap Sort, Red-black Trees		
39		Red-black Trees		
40		**Ticking Session**		
41		2-3Trees		
42		B Trees :Key Insertion, Key Search		
43		B Trees: Key Deletion		
44		**Ticking Session**		
45		**ISA 3**		
46	**Unit #4:** **T1:** **chapters7.1,7** **.2,** **9.1,9.2,9.3,9.4**	Space and Time Tradeoffs-Sorting by Counting	**17**	**77**
47		Distribution Counting Sort		
48		Input Enhancement: String Matching Horspool's algorithm Practice: Horspool Algorithm implementation		
49		Boyer-Moore Algorithm		
50		**Ticking Session**		
51		Greedy Technique		
52		Prim's Algorithm		
53		Kruskal's Algorithm and union and find algorithm **Practice:** Minimum Spanning Tree implementation		
54		Dijkstra's Algorithm **Practice:** Dijkstra's algorithm		
55		**Ticking Session**		
56		Huffmantrees		
57		Huffman Tree implementation		
58		**ISA-4**		
59	**Unit 5:** **Chapters: 8.1, 8.2, 8.4, 11.1,11.2, 11.3, 12.1,12.2**	Limitations of Algorithm Power :Lower-Bound Arguments	23	100
60		Decision Trees		
61		P,NP, and NP-Complete NP-Hard Problems		
62		Coping with the Limitations of Algorithm Power		
63		**Ticking Session**		
64		Backtracking **Practice:** N queens using backtracking		
65		Branch-and-Bound		
66		Dynamic Programming :Computing a Binomial Coefficient **Practice:** Implementation of Binomial coefficient using dynamic programming		
67		The Knapsack Problem solutions using		

		Dynamic Programming		
68		**Ticking Session**		
69		Memory Functions for solving Knapsack Problem **Practice:** Implementation of Memory Function for Solving Knapsack Problem		
70		Warshall’s Algorithmt of in dTransitive Closure		
71		Floyd’s Algorithm for All Pair Shortest path problem **Practice:** Implementation of all pair shortest path		
72		**Ticking Session**		
73		**ISA-5**		
74		Assignment / Hackathon		
75		Assignment / Hackathon		

Ticking Session:

1. Ticking Sessions are typically one hour classroom session where students engage in solving problems
2. These problems kindle student logical thinking, problem solving and programming skills
3. Students are expected to get all the ticks for the problems that is given to them by the end of course
4. A typical Ticking session will consist of 2-4 problems depending on the complexity of the problem
5. The problems will be of similar type but not the same problems across various sections. Faculty can choose a few questions of their choice and complexity from the pool of questions.
6. Faculty are to encourage students to get all the ticks during the classroom sessions
7. Every tick the student get in classroom session will be a **green tick**(full 5 marks). Every tick the student gets outside the classroom session will be **orange tick**(partial 3 marks). Every tick that the student do not complete will be **red tick**(no marks)

Each question:

1. Story describing what to be done
2. Sample input
3. Expected output
4. Skeleton code / stub code
5. 3 – 4 test cases, should cover all possible cases
6. Put this in sublit
7. Students can submit

BookType	Code	Title&Author	PublicationInformation		
			Edition	Publisher	Year
Text Book	T1	Introduction to The Design and Analysis of Algorithms Anany Levitin	2	Pearson	2012
Referenc eBook	R1	Introduction to Algorithms ThomasH.Cormen,CharlesE.Leiserson,RonaldL.RivestandCliffordStein	3	Prentice-HallIndia	2009
Referenc eBook	R2	Fundamentals of Computer Algorithms Horowitz, Sahni, Rajasekaran,	2	UniversitiesPress	2007
Reference Book	R3	Algorithm Design JonKleinberg,EvaTardos,	1	Pearson Education	2006

UE21CS242B: Operating Systems (4-0-1-4-4)

No. of Credits: 4 **# of Hours: 75**

Class #	Chapter Title/Reference Literature	Topics to be covered	Reference	% of Portions Covered	
				Reference Chapter	Cumulative
1	Unit: 1 Introduction and Process Management	What Operating Systems Do, Computer-System Organization	T1: 1.1 - 1.2	23	23
2		Computer-System Architecture, Operating-System Structure & Operations	T1: 1.3 - 1.5		
3		Kernel Data Structures, Computing Environments	T1: 1.10 - 1.11		
4		Operating-System Services, Operating System Design and Implementation	T1: 2.1 - 2.6		
5		Lab 1 - Creation of Linux VM, installation of C compiler, creation of a sample program, Linux shells, basic Linux commands			
6		Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	T1: 3.1 - 3.3		
7					
8		System calls for process management	T2: 8.1 – 8.10		
9		Lab 2 - Demonstration of process management system calls			
10		CPU Scheduling: Basic Concepts, Scheduling Criteria	T1: 6.1, 6.2		
11		Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	T1: 6.3		
12		Scheduling Algorithms: Priority Scheduling, Round-Robin Scheduling	T1: 6.3		
13		Lab 3 - Demonstration of process scheduling algorithms			
14		Multi-level Queue, Multi-Level Feedback Queue Scheduling	T1: 6.3		
15		Case Study: Linux Scheduling	T1: 6.7		
16		Programming exercise on process management			
17		ISA 1			

18	Unit : 2 IPC, Threads and Concurrency	IPC: Shared Memory & MessagePassing, Pipes-Named and Ordinary	T1: 3.4, 3.6	21	44
19		System calls for shared memory, pipes and FIFOs	T2: 15		
20		Lab 4 - Demonstration of shared memory, pipes and FIFOs system calls			
21		Introduction to Threads, types of threads, Multicore Programming.	T1: 4.1, 4.2		
22		Multithreading Models, Thread creation, Thread Scheduling	T1: 4.3, 6.4		
23		Thread libraries, Pthreads and Windows Threads	T1: 4.4		
24		Lab 5 - Demonstration of threads			
25		Mutual Exclusion and Synchronization: software approaches	T1: 5.1-5.3		
26		Principles of concurrency, hardware support	T1: 5.4		
27		Mutex Locks, Semaphores	T1: 5.5-5.6		
28		Classic problems of Synchronization: Bounded-Buffer Problem, Readers - Writers problem, Dining Philosophers Problem concepts	T1: 5.7		
29		Synchronization Examples	T1: 5.9		
30		Deadlocks: principles of deadlock, Deadlock Characterization	T1: 7.1, 7.2		
31					
32		Lab 6 – Demonstration of mutex, semaphores, deadlocks			
33		Programming exercise on interprocess communication			
34		ISA 2			
35	Unit :3 Memory Management	Main Memory: Hardware and control structures, OS support, Address translation, Dynamic Loading, Dynamic Linking and Shared Libraries	T1: 8.1	20	64
36		Swapping, Memory Allocation (Partitioning, relocation), Fragmentation	T1: 8.2-8.3		
37		Segmentation	T1: 8.4		
38		Paging	T1: 8.5		
39		Structure of page tables	T1: 8.6		

40		Example: Intel 32 and 64-bit Architectures	T1: 8.7		
41		Virtual Memory – Demand Paging,	T1: 9.1, 9.2		
42		Copy-on-Write	T1: 9.3		
43		Page replacement	T1: 9.4		
44		Allocation of frames	T1: 9.5		
45		Thrashing	T1: 9.6		
46		Case Study: Linux/Windows Memory	T1: 9.10		
47		Lab 7 – Demonstration of page replacement algorithms			
48		Programming exercise on virtual memory management			
49		ISA 3			
50	Unit : 4 File Management	File Concept	T1: 11.1, 11.2	17	81
51		Access Methods, Directory and Disk Structure	T1: 11.3		
52		File-System, sharing, File system protection	T1: 11.4, 11.6		
53		System calls to retrieve file attributes, file types and file operations	T2: 4		
54		System calls for reading directories, create hard links and symbolic links	T2: 4		
55		File system implementation	T1: 12.1-12.2		
56		Directory implementation, allocation methods	T1: 12.3-12.4		
57		Free-Space Management	T1: 12.5		
58		Efficiency and Performance	T1: 12.6		
59		Case study: Linux	T1: 12.9		
60		Lab 8 - Demo of file operations			
61		Programming exercise on file management			
62		ISA 4			
63	Unit : 5 Storage Management and System Protection	Mass storage structure, Disk Structure	T1: 10.1-10.3	19	100
64		Disk scheduling, Disk Management	T1: 10.4-10.5		
65		Swap-Space Management,	T1: 10.6		
66		RAID Structure	T1: 10.7		
67		System Protection: Goals, Principles and Domain of Protection	T1: 14.1-14.3		

68		Access Matrix, Implementation of the Access Matrix	T1: 14.4 - 14.5		
69		Access Control, Revocation of Access Rights	T1: 14.6- 14.7		
70		System calls for access control	T2: 6		
71		Case Study: Windows, Linux			
72		Lab 9 - Demonstration of access control			
73		Programming exercise on access control			
74		Project review			
75		ISA 5			

Tools/ Languages/OS : Pthreads, C, Linux/Unix OS for system call implementation.

Text Book(s):

1. "Operating System Concepts", Abraham Silberschatz, Peter Baer Galvin, Greg Gagne 9th Edition, John Wiley & Sons, India Edition ,2016.
2. "Advanced Programming in the Unix Environment", Richard Stevens and Stephen A Rago, Pearson, 3rd edition, 2017.

Reference Book(s):

1. "Operating Systems, Internals and Design Principles", William Stallings, 9th Edition, Pearson, 2018.
2. "Operating Systems": Three Easy Pieces, RemziArpaci-Dusseau and Andrea Arpaci Dusseau, http://pages.cs.wisc.edu/~remzi/OSTEP/.
3. "Operating Systems", Harvey Deitel, Paul Deitel, David Choffnes, 3rd Edition, Prentice Hall, 2004.
4. "Modern Operating Systems", Andrew S Tanenbaum, 3rd edition, Pearson, 2007.

UE21MA241B: Linear Algebra (4-0-1-4-4)

No. of Credits: 4 **# of Hours: 75**

Class No.	Chapter Title / Reference	Portions to be Covered	Percentage of Syllabus Covered	Percentage of SyllabusCovered (Cumulativ)
1	Unit 1:Matrices and Gaussian Elimination T1:pg :3-9, Pg: 11-15, Pg:21-22, pg: 32-39, pg :45-47, pg: 49-51	Introduction to Linear Algebra	20	20
2-3		The Geometry of Linear Equations – Row and Column Pictures		
4		Singular cases in two and three dimensions		
5-7		Gaussian Elimination, The breakdown of elimination		
8		Elimination Matrices		
9-10		Triangular Factors (LU and Cholesky method) and Row Exchanges		
11		Inverse by Gauss -Jordan Method, Transposes		
12		Applications		
13		Matlab Class Number1 – Introduction		
14		Classwork/Assignment		
15		ISA1		
16-17	Unit 2: Vector Spaces T1: pg 69-71, pg 78-81,pg 92-98, pg 102-105, R2: 7.6 R1:Pg 178-179	Vector Spaces and Subspaces (Definition only), Column Space and Null Space, Examples	20	40
18-19		Echelon Form, Row Reduced Form, Pivot Variables , Free variables		
20		Sum of subspaces, Direct sums		
21-23		Linear Independence, Basis and Dimensions		
24		The Four Fundamental Subspaces		
25		Uniqueness and Existence of Inverses Rank Nullity theorem		
26		Applications		
27		Matlab Class Number 2 – Gaussian Elimination		
28		Matlab Class Number 3 -Inverse of a Matrix by Gauss Jordan Method		
29		Classwork/Assignment		
30		ISA2		
31, 32	Unit 3:	Linear Transformations , Examples, Transformations Represented by Matrices		
33		Algebra of Linear Transformations, Invertible maps, Isomorphisms		
34, 35		Rotations, Reflections and Projections		
36,		Orthogonal Vectors and Subspaces, Orthogonal		

37	Orthogonality T1:pg 125-127, pg 127-129,pg 130-132,pg 141-148,pg 152-157,pg 160-167 R2: 11.1 to 11.4 R1:Pg 250-251	Bases	20	60
38, 39		Cosines and Projections onto Lines		
40		Projections and Least Squares		
41		Applications		
42		Classwork/Assignment		
43		Matlab Class Number 4 – LU Decomposition		
44		Matlab Class Number 5,6 - Span of Column Space of A, Four Fundamental Subspaces of A		
45		ISA3		
46-48	Unit 4: Orthogonalization, Eigen Values and Eigen Vectors T1:pg 174-178,pg 179-182, R3: pg 55-60, R1: pg 465-468 T1: pg 245-247, T1: pg 248-249,Pg 285 R1: Pg 452	Orthogonalization - Orthogonal Matrices, Properties, Rectangular Matrices with orthonormal columns	20	80
49-50		The Gram- Schmidt Orthogonalization, A = QR Factorization		
51-53		Introduction to Eigen values and Eigenvectors, Properties of eigenvalues and Eigenvectors, Spectral theorem, Symmetric Matrices, Cayley-Hamilton theorem(Statement only)		
54, 55		Diagonalization of a Matrix, Powers and Products of Matrices		
56		Applications		
57		Matlab Class Number 7 - Projections by Least Squares		
58		Matlab Class Number 8 - The Gram- Schmidt process.		
59		Classwork/Assignment		
60		ISA4Matlab Class Number 9 - QR Factorization Matlab Class Number 10 - Eigen Values and Eigen Vectors		
61	Unit 5: Singular Value Decomposition R1: pg 471-472, 477 T1: pg 318-319 T1: pg 319-321 T1: pg 331-332 R1: pg 487-494 R1: pg 500-501 T1:Pg 335-336	Matlab Class Number 9 - QR Factorization	20	100
62		Matlab Class Number 10 - Eigen Values and Eigen Vectors		
63-65		Quadratic Forms, Definitions of positive definite, negative definite, positive semi-definite, negative semi-definite, Indefinite forms and Matrices		
66		Tests for Positive Definiteness		
67-68		Problems on Positive Definite Matrices and Least Squares, Problems on Semi-definite Matrices		
69-71		The Singular Value Decomposition of a Matrix, Examples		
72		Pseudoinverse		
73		Applications		
74		Matlab - In Semester Assessment		
75		ISA5		

Text Book:

T1: "Linear Algebra and its Applications", Gilbert Strang, 4th Edition, Thomson Brooks/ Cole, Second Indian Reprint 2007..

Reference Books:

R1: Linear Algebra and its Applications, David .C lay, Publication by Pearson, 5th Edition, 2015.
R2: Linear Algebra, Schaum's outlines, Seymour Lipschutz and Marc Lipson, 4th Edition, McGraw-Hill publications, 2009.
R3**:** Higher Engineering Mathematics, B S Grewal, 44th Edition, Khanna Publishers,2020.
R4: Practical Linear Algebra, Gerald Farin and Dianne Hansford, 3rd Edition, CRC Press, Taylor & Francis Group, 2013.

Reference Books:

1: Getting started with MATLAB, Rudra Pratap, Oxford University Press, 7th Edition, 2016.
2: MATLAB for Engineers, Holly Moore, Pearson Publications, New Jersey, 5th Edition, 2018.