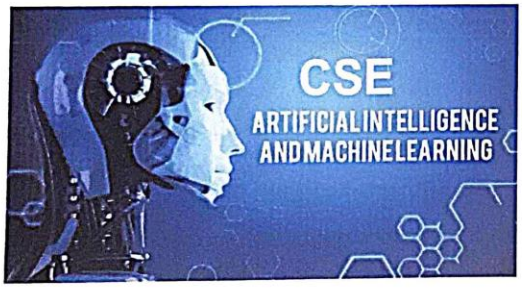
SCHEME AND SYLLABUS



2022 SCHEME

Computer Science and Engineering (AI&ML)

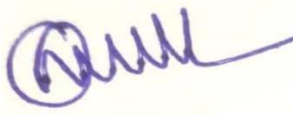
III – IV Semester

GLOBAL ACADEMY OF TECHNOLOGY
(Autonomous Institution Affiliated to VTU, Belagavi.)
Accredited by NAAC with 'A' Grade,
NBA Accredited - CSE, ISE, ECE, EEE, ME, CV
Ideal Homes Township,
Raja Rajeshwari Nagar, Bengaluru-560098.

Dean Academic
Global Academy of Technology,
Raja... hwarinagar, Bengal...-98

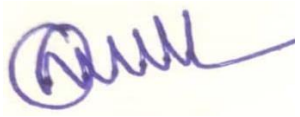
SEMESTER III

Sl. No.	Course Code	Course Title	Course Type	Teaching Dept.	Teaching Hours/Week			Examination			CREDITS
					L	T	P	CIE	SEE	Total	
1	22MAT31A	Discrete Mathematics and Graph Theory	BS	MAT	2	2	0	50	50	100	3
2	22CML32	Data Structures (Integrated)	IPC	Respective Department	3	0	2	50	50	100	4
3	22CML33	Digital Design and Computer Organization (Integrated)	IPC		3	0	2	50	50	100	4
4	22CML34	Operating Systems	PC		3	0	0	50	50	100	3
5	22CML35	Introduction to Web Technologies(Integrated)	PLC		2	0	2	50	50	100	3
6	22CML36	Unix and Shell Programming (Integrated)	AEC		2	0	2	50	50	100	3
				Total				**300**	**300**	**600**	**20**



SEMESTER IV

Sl. No.	Course Code	Course Title	Course Type	Teaching Dept.	Teaching Hours/Week			Examination			CREDITS
					L	T	P	CIE	SEE	Total	
1	22MAT41A	Probability and Linear Algebra	BS	MAT	2	2	0	50	50	100	3
2	22CML42	Database Management Systems (Integrated)	IPC	Respective Department	3	0	2	50	50	100	4
3	22CML43	Design and Analysis of Algorithms (Integrated)	IPC		3	0	2	50	50	100	4
4	22CML44	Software Engineering	PC		3	0	0	50	50	100	3
5	22CML45	Object Oriented Programming with Java (Integrated)	ETC		2	0	2	50	50	100	3
6	22CML46	Advanced Python	AEC		2	2	0	50	50	100	3
Total								**300**	**300**	**600**	**20**



SEMESTER – III

Course: Discrete Mathematics and Graph Theory
(Common for CSE/ISE/AI&DS /AI&ML/AI&ML/CS(AIML))

Course Code	**22MAT31A**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:2:0**	**SEE Marks**	**50**
No. of Credits	**3**	**Examination Hours**	**03**

Course Objectives: To enable students to apply the knowledge of Mathematics in fields of computer science and allied branches by making them to learn:

CLO1	Counting Principles
CLO2	Mathematical Logic and Set Theory
CLO3	Relations and Functions
CLO4	Graph Theory

Content	No. of Hours/ RBT levels
Module 1 The Rules of Sum and Product, The Pigeon-hole Principle, Permutations, Combinations, The Binomial Theorem, Combinations with Repetition.	**08 Hours** **L2, L3**
Module 2 Sets and Subsets, Set operations and Laws of Set Theory. Counting and Venn Diagrams. Probability, Conditional probability and Bayes Theorem.	**08 Hours** **L2, L3**
Module 3 Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implications: Rules of Inference. Quantifiers, Definitions and the Proofs of Theorems.	**08 Hours** **L2, L3**
Module 4 Relations and properties of relations, Representation of relations. Equivalence Relations and Partitions. Functions, Types of Functions, Function Composition and Inverse Functions.	**08 Hours** **L2, L3**
Module 5 Graphs, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits. Planar Graphs, Trees and Rooted Trees.	**08 Hours** **L2, L3**

COURSE OUTCOMES:

Upon completion of this course, student will be able to:

CO31.1	Use computational techniques essential for the study of mathematical logic, set operations, relations and functions.
CO31.2	Solve problems using basic graph theory

Textbooks:

1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education. 2020.

Reference books:

1. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007

Scheme of Examination:

Semester End Examination (SEE):
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of three sub questions) from each module carrying 20 marks each. Students are required to answer any **five full questions** choosing at least **one full question from each module.**

Continuous Internal Evaluation (CIE):
Three Tests are to be conducted for 40 marks each. Average of Marks scored in all three tests is added to test component. CIE is executed by way of quizzes / Alternate Assessment Tools (AATs), and three tests. **Some possible AATs:** seminar/assignments/ mini-projects/ concept videos/ partial reproduction of research work/ group activity/ any other.

Typical Evaluation pattern for regular courses is shown in Table 2.

Table 2: Distribution of weightage for CIE & SEE of Regular courses

Component		Marks	Total Marks
CIE	CIE Test-1	**40**	**50**
	CIE Test-2	**40**	
	CIE Test-3	**40**	
	Assignments	**10**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

CO/PO Mapping																
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO31.1	3	2	1									3				
CO31.2	3	2	1									3				
Average	3	2	1									3				

Low-1: Medium-2: High-3

SUBJECT: DATA STRUCTURES (Integrated)

Subject Code	**22CML32**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:2**	**SEE Marks**	**50**
Total Hours	**50**	**Examination Hours**	**3**
No. of Credits: 04			

Course Learning Objectives:

The course will enable students to:

CLO1	Discuss the classification of data structures, arrays, pointers, dynamic memory allocation
CLO2	Explain the working of Stacks, Queues and their operations
CLO3	Apply the concepts of singly linked lists, doubly linked lists, circular linked lists to solve a given Problem
CLO4	Explore usage of Trees and Graph for application development
CLO5	Apply the Hashing techniques in mapping key value pairs.

CONTENTS	No. of Hours / RBT Levels
MODULE 1 **Introduction:** Data Structures, Classifications of Data Structures, Data structure Operations, Review of Arrays, type of Structures, Self-Referential Structures, and Unions. Review of Pointers, Dynamic Memory Allocation Functions, Dynamically allocated arrays **Array Operations**: Traversing, inserting, deleting, searching, and sorting, Selection Sort, Insertion Sort, Multidimensional arrays, Sparse matrices, and transpose of sparse Matrices **Text Book: T1 Chapter: 2 Text Book: T2 Chapters:1, 4**	**10** **L3**
MODULE 2 **Stacks:** Definition, Stack Operations, Array Representation of Stacks, Polish notation **Applications of stack:** Infix to postfix conversion, evaluation of postfix expression, Recursion- Fibonacci Sequence, Tower of Hanoi, Ackermann Functions **Queues:** Definition, Queue Operations, Array Representation of Queues, Circular Queues and operations, Priority Queue **Text Book: T1 Chapter: 3 Text Book: T2 Chapter: 6**	**10** **L3**
MODULE 3 **Linked Lists:** Definition, Representation of linked lists in Memory, Linked list operations: Traversing, Searching, Insertion, and Deletion. Stack operation using linkedlist and queue operation using linked list. Doubly Linked lists, Circular linked lists, and header linked lists, Applications of Linked lists –Polynomial Representation and Addition of Polynomial **Text Book: T1 Chapter: 4 Text Book: T2 Chapter: 5**	**10** **L3**
MODULE 4 **Trees:** Terminology, Binary Trees, Properties of Binary Trees, Array and Linked representation of Binary Trees, Types of Binary tree, Construction of Binary Tree, Binary Tree Traversals - Inorder, Postorder, Preorder **Binary Search Trees:** Definition, Construction of Binary Search Tree, Insertion,	**10** **L3**

Deletion, Traversal and Searching **Text Book: T1 Chapter: 5 Text Book: T2 Chapters: 7**	
MODULE 5 Construction and Evaluation of Expression Trees, Threaded binary trees **Hashing:** Hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing **Sorting:** Radix sort, Address Calculation Sort **Text Book: T1 Chapter: 5 Text Book: T2 Chapter:9**	**10** **L3**

Laboratory Component

List of Experiments

Implement the following programs using C in Ubuntu

1. Design, Develop and Implement a menu driven Program for the following Array operations
   a. Creating an Array of N Integer Elements
   b. Display of Array Elements with suitable headings
   c. Inserting an element (ELEM) at a given valid position (POS)
   d. Deleting an element at a given valid position (POS)
   e. Exit
   Support the program with functions for each of the above operations.
2. Design, Develop and Implement a program to sort the given list of 'n' integers in increasing/decreasing order using Insertion sort algorithm
3. Design, Develop and Implement a menu driven Program for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
   a. Push an Element on to Stack
   b. Pop an Element from Stack
   c. Demonstrate Overflow and Underflow situations on Stack
   d. Display the status of Stack
   e. Exit
   Support the program with appropriate functions for each of the above operations
4. Design, Develop and Implement a Program for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, %(Remainder), ^(Power) and alphanumeric operands.
5. Design, Develop and Implement a Program for evaluation of Stack Suffix expression with single digit operands and operators: **+, -, *, /, %,** ^
6. Design, Develop and Implement a menu driven Program for the following operations on Ordinary QUEUE of Characters (Array Implementation of Queue with maximum size MAX)
   a. Insert an Element on to Ordinary QUEUE
   b. Delete an Element from Ordinary QUEUE
   c. Demonstrate Overflow and Underflow situations on Ordinary QUEUE
   d. Display the status of Ordinary QUEUE
   e. Exit
   Support the program with appropriate functions for each of the above operations.
7. Design, Develop and Implement a menu driven Program for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to Circular QUEUE
b. Delete an Element from Circular QUEUE
c. Demonstrate Overflow and Underflow situations on Circular QUEUE
d. Display the status of Circular QUEUE
e. Exit

Support the program with appropriate functions for each of the above operations.

8. Design, Develop and Implement a menu driven Program for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo
   a. Create a SLL of N Students Data by using front insertion.
   b. Display the status of SLL and count the number of nodes in it
   c. Perform Insertion / Deletion at End of SLL
   d. Exit
9. Design, Develop and Implement a menu driven Program for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: emp_id, Name, Dept, Designation, Sal.
   a. Create a DLL of N Employees Data by using end insertion.
   b. Display the status of DLL and count the number of nodes in it
   c. Perform Insertion and Deletion at Front of DLL
   d. Demonstrate how this DLL can be used as Double Ended Queue
   e. Exit
10. Develop a C program for performing the following operations on Binary Search Tree (BST) of Integers
    a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
    b. Traverse the BST in Inorder, Preorder and Post Order
    c. Search the BST for a given element (KEY) and report the appropriate message.
    d. Exit

Course Outcomes:

Upon successful completion of this course, student will be able to

CO32.1	Apply array concepts to sort the elements, transpose the sparse matrix
CO32.2	Develop programs on operations of stack and its applications, recursion, queue operations
CO32.3	Apply the concepts of singly linked lists, doubly linked lists, circular linked lists into different applications
CO32.4	Illustrate the construction of binary trees, binary search tress, and its traversal techniques
CO32.5	Implement the algorithms related to expression tree, hashing and hashing-based sorting techniques

Text Books:

1. Ellis Horowitz, Sartaz Sahni, Anderson, Freed, "Fundamentals of Data Structures in C", 2nd Edition, University Press, 2008, Reprinted 2016.
2. Lipschutz, Schaum's Outlines, "Data Structures using C", Seymour McGraw Hill Special Indian Edition, 13th Reprint 2015.

Reference Books:

1. Yedidyah Langsam, Moshe J Augenstein and Aaron M Tanenbaum, "Data Structures using C & C++", 2nd Edition, Pearson, 10th Impression 2020.
2. Richard F. Gilberg and Behrouz A. Forouzan, "Data Structures - A Pseudo code approach with C", 2nd Edition, Cengage Learning, Sixth Indian Reprint 2016.

E-Books / Web References:

1. https://www.freebookcentre.net/ComputerScience-Books-Download/Data-Structures-and-Algorithms.html
2. http://www.uoitc.edu.iq/images/documents/informatics-institute/Competitive_exam /DataStructures.pdf
3. https://people.cs.vt.edu/shaffer/Book/Java3e20110103.pdf

MOOCs:

1.https://www.edx.org/course/introduction-to-data structures

2. https://nptel.ac.in/courses/106/102/106102064/

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO32.1	3	3	2	-	-	-	-	-	1	1	-	2	3	2
CO32.2	3	3	2	-	-	-	-	-	1	1	-	2	3	2
CO32.3	3	3	2	-	-	-	-	-	1	1	-	2	3	2
CO32.4	3	3	2	-	-	-	-	-	1	1	-	2	3	2
CO32.5	3	3	2	-	-	-	-	-	1	1	-	2	3	2
Average	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**1**	**1**	**-**	**2**	**3**	2

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

SUBJECT: DIGITAL DESIGN AND COMPUTER ORGANIZATION (Integrated)

Subject Code	**22CML33**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:2**	**SEE Marks**	**50**
Total Hours	**50**	**Examination Hours**	**3**
No. of Credits: 04			

Course Learning Objectives:
The course will enable students to:

CLO1	Understand the basic digital principles and working of various logic gates, and different techniques for simplification of Boolean function
CLO2	Design combinational logic circuits and describe their applications
CLO3	Understand the working of Flip-Flops and Counters
CLO4	Understand the basic sub systems of a computer, their organization, structure and operation
CLO5	Learn arithmetic and logical operations with integer and floating-point operands

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Combinational Logic Circuits:** Boolean functions, The Map Method: Two, Three, Four Variable Maps, Map Manipulation, Product-of-Sums, Sum-of-Products Optimization, Don't Care Conditions, The Tabulation Method, Determination of Prime Implicants **Text Book: T1 Chapters: 4, 5, 6**	**10** **L3**
MODULE 2 **Data Processing Circuits:** Adders, Subtractors, Code Converters, Magnitude Comparators, Multiplexers, Demultiplexers, Encoders, Decoders, Programmable Array Logic, Programmable Logic Arrays **Text Book:T1 Chapters: 9**	**10** **L3**
MODULE 3 **Sequential Logic**: Introduction to Flip-Flops, Types of Flip flops, Various Representation of Flip-Flops, JK Master-Slave Flip-Flop, Flip-Flop Turning, Switch Contact Bounce Circuits **Registers:** Types of Registers. Serial In - Serial Out, Serial In - Parallel out, Parallel ln - Serial Out, Parallel ln - Parallel Out, Applications of Shift Registers **Counters:** Asynchronous Counters, Synchronous Counters, Counter Design as a Synthesis problem **Text Book:T1 Chapter: 11,12**	**10** **L3**
MODULE 4 **Machine Instructions and Programs:** Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language **Input/Output Organization:** Basic Input and Output Operations, Accessing IO Devices, Interrupts - Interrupt Hardware, Enabling and Disabling Interrupts, Direct Memory Access, Buses **Text Book: T2 Chapters: 2, 4**	**10** **L2**

MODULE 5	
Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication- Bit-pair recoding of multipliers, Integer Division, Floating-point Numbers and Operations **Text Book: T2 Chapter: 6**	**10** **L3**

Laboratory Component

Hardware experiments to be conducted using logic trainer kits and simulation experiments to be conducted using open-source software like Logisim.

List of Experiments

1. a. Study and verify the truth tables of AND, OR, NOT, NOR, NAND and XOR Logic Gates.
   b. Simplify the given Boolean expression and realize it using Basic gates and Universal Gates.
2. Design and implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using Logic Gates. Also simulate their working.
3. Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC.
4. Design and implement 4-bit Parity checker.
5. Design and implement Code converter i) Binary to Gray ii) Gray to Binary Code using Logic gates.
6. Realize a J-K Master Slave flip-flop using NAND gates and verify its truth table.
7. Design and implement 4-bit Ring and Johnson Counters. Also simulate their working.
8. Design and implement mod-n (n<8) synchronous up counter using JK flip-flop ICs and demonstrate its working.
9. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9).
10. Design and implement a pseudo-random sequence generator using shift register IC7495.

Demonstration using of virtual lab

https://www.vlab.co.in/

- Representation of Integers and their Arithmetic
- Floating Point Numbers Representation

Course Outcomes:

Upon successful completion of this course, student will be able to

CO33.1	Illustrate the minimization of combinational logic expressions using K-map and Quine McCuskey methods
CO33.2	Interpret different combinational logic circuits like Adders, Subtractors, Multiplexers, Decoders and programmable Logic Arrays
CO33.3	Implement the Flip Flops, Registers and Counters
CO33.4	Explain the basic structure of computers, machine instructions & addressing modes
CO33.5	Solve Arithmetic operations on binary numbers

Text Books:

1. Charles H. Roth, Jr. and Larry L Kinney, "Fundamentals of Logic Design", 6th Edition, Thomson, 2010.
2. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", McGraw Hill Education India, 5th Edition, 2011.

Reference Books:

1. Neal S Widmer, Greg Moss and Ronald J Tocci, "Digital Systems Principles and Applications",12th Edition, 2022, Pearson.
2. William Stallings, "Computer Organization & Architecture", 9th Edition, Pearson. 2015.

E-Books / Web References:

1. Computer Organization & Architecture - William Stallings, http://home.ustc.edu.cnl-louwenqi/reference-books-%20WiIIiarm%20StaIIings.pdf

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO33.1	3	2	2	1	-	-	-	-	-	1	-	-	1	
CO33.2	3	2	2	-	2	-	-	-	-	1	-	-	1	
CO33.3	3	2	2	-	2	-	-	-	-	1	-	-	1	
CO33.4	2	1	-	-	-	-	-	-	-	-	-	-	1	
CO33.5	2	1	2	-	-	-	-	-	-	-	-	-	1	
Average	**2**	**1**	**2**	**1**	**2**	**-**	**-**	**-**	**-**	**1**	**-**	**-**	**1**	

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

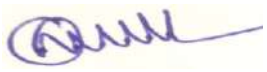
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**



SUBJECT: OPERATING SYSTEMS

Subject Code	**22CML34**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:0**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**03**
No. of Credits: 3			

Course Learning Objectives:

The course will enable students to

CLO1	Understand the fundamentals of an Operating Systems and its structures, concept of processes and threads
CLO2	Understand the implement efficient Process scheduling mechanisms and software solutions for process synchronization
CLO3	Know Deadlock handling mechanism
CLO4	Understand Operations in Memory Management
CLO5	Study the Virtual memory and mass storage Concepts

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction to operating System:** What operating system do, Operating System Services, Systems Calls, **Processes:** Process Concepts, Process Scheduling, Inter Process Communication, Thread Overview, Multicore Programming, Multithreading Models **Text book:T1 Chapters: 1.1, 2.1, 2.3, 3.1, 3.2, 3.4, 4.1, 4.2, 4.3**	**08** **L2**
MODULE 2 **Process Scheduling:** CPU Scheduling - Basic concepts, Scheduling Criteria, Scheduling Algorithms **Process Synchronization:** Background, Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores. Classic problems of Synchronization **Text book: T1 Chapters: 5.1, 5.2, 5.3, 6.1, 6.2, 6.3, 6.5, 6.6, 7.1**	**10** **L3**
MODULE 3 **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention. Deadlock Avoidance: Banker's Algorithm, Deadlock Detection and recovery from Deadlock **Text book:T1 Chapters: 8.1, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8**	**07** **L3**
MODULE 4 **Memory management:** Background**,** Contiguous memory allocation, paging, Structure of page table, Swapping **Text book:T1 Chapters: 9.1, 9.2, 9.3, 9.4, 9.5**	**08** **L2**
MODULE 5 **Virtual memory management** Background, Demand Paging, Page Replacement algorithms: FIFO page replacement, Optimal page replacement, LRU page replacement, thrashing: Cause of Thrashing **Secondary Storage Structures**	**07** **L3**

HDD Scheduling, Storage Device Management, RAID Structure-Redundancy, Parallelism, RAID levels, Selecting a RAID Level **Text book: T1 Chapters:10.1, 10.2, 10.4.1, 10.4.2, 10.4.3, 10.4.4, 10.6.1, 11.2, 11.5, 11.8.1, 11.8.2, 11.8.3, 11.8.4.**	

Course Outcomes:

Upon successful completion of this course, student will be able to

CO34.1	Understand the basic concepts of operating systems and concept of processes and threads
CO34.2	Explain the process management, CPU scheduling and synchronization tools
CO34.3	Explain the deadlock handling methods
CO34.4	Describe memory management mechanisms
CO34.5	Illustrate Virtual memory concepts and mass storage structure

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles, 10th Edition, Wiley-India, 2018.

Reference Books:

1. Stallings, William, Operating systems: Internals and design principles. Prentice Hall Press, 2011.
2. Andrew S Tanenbaum and Herbert Bos, Modern Operating Systems, 4th Edition, Pearson Education, 2014.
3. Thomas Anderson and Michael Dahlin, Operating Systems: Principles and Practice, Recursive Books, 2014.
4. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice, 4th Edition, PHI, 2014.

E-Books / Web References

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. Operating System Concepts, Wiley India, 10th Edition, 2018.
2. Andrew S. Tanenbaum, Modern Operating Systems, PHI, 3rd Edition, 2009.
3. http://edclap.com/mod/resource/view.php?id=1445&forceview=1
4. Operating Systems Course Notes: by Dr. John T.Bell https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/

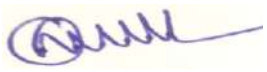
MOOCs

1. http://onlinevideolecture.com/?course=computer-science&subject=operating-systems
2. https://nptel.ac.in/courses/106/106/106106144/
3. http://www.nptel.ac.in/courses/106108101/

Mapping of CO-PO

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO34.1	3	3	3	-	-	-	-	-	-	-	-	3	2	-
CO34.2	3	3	3	-	-	-	-	-	-	-	-	3	2	-
CO34.3	3	3	3	-	-	-	-	-	-	-	-	3	2	-
CO34.4	3	3	3	-	-	-	-	-	-	-	-	3	2	-
CO34.5	3	3	3	-	-	-	-	-	-	-	-	3	2	-
Average	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	2	**-**

Low-1: Medium-2: High-3



Scheme of Evaluation:

Semester End Examination (SEE):

SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three tests are to be conducted for 40 marks each. Average of all three tests marks is added to the test component. In order to encourage innovative methods while delivering a course, the faculty members have been encouraged to use the Alternative Assessment Tool (AAT). The AAT enhances the autonomy (freedom and flexibility) of individual faculty and enables them to create innovative pedagogical practices. Possible AATs are – seminar / assignments / term paper / open ended experiments / mini-projects/group activity or any other.

Table: Distribution of weightage for CIE & SEE

	Component	Marks	Total Marks
CIE	CIE Test-1	**40**	**50**
	CIE Test-2	**40**	
	CIE Test-3	**40**	
	AAT	**5**	
	AAT	**5**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

SUBJECT: INTRODUCTION TO WEB TECHNOLOGIES(Integrated)

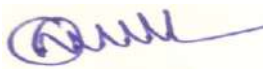
Subject Code	**22CML35**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:0:2**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**3**
No. of Credits: 3			

Course Learning Objectives:

The course will enable students to:

CLO1	Illustrate the Semantic Structure of HTML and CSS
CLO2	Compose forms and tables using HTML and CSS
CLO3	Understand different approaches for creating page layouts
CLO4	To build dynamic web pages using JavaScript.
CLO5	Understand Responsive Web Design with the help of the Bootstrap framework

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction to HTML:** What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements. **Introduction to CSS**: What is CSS, CSS Syntax, Location of Styles, Selectors **Text book:1 Chapter 2, 3 (till selectors)**	**8** **L3**
MODULE 2 **The Cascade**: How Styles Interact, The Box Model, CSS Text Styling. **HTML Tables and Forms:** Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats **Text book:1 Chapter 3, 4**	**8** **L3**
MODULE 3 **Advanced CSS:** Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks **Text book:1 Chapter 5**	**8** **L3**
MODULE 4 **JavaScript:** Client-Side Scripting, What is JavaScript and What can it do?, JavaScript Design Principles, Where does JavaScript Go?, Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms, Examples **Text book:1 Chapter 6**	**8** **L3**
MODULE 5 **Bootstrap:** What is bootstrap?, Bootstrap File structure,Default Grid system, Fluid Grid system,Container Layouts; **Bootstrap CSS:** Typography, Tables,Forms, Buttons, Images,icons; **Bootstrap Layout components:** Dropdown Menus, Button	**8** **L3**



Groups,Buttons with dropdowns,Navigation elements, Navbar, Breadcrumps, Pagination, Typographic elements **Text book: 2 Chapter 1,2,3**	

Laboratory Component

List of Experiments

1. Develop a webpage that gives information about travel experience using the following HTML5 Semantic tags- <article>, <aside>, <figcaption>,<figure>, <footer>, <header>, <main>, <nav>, <section>
2. Build a clean and organized layout for the webpage using HTML.
   a. Include a header section with the event title, date, and location.
   b. Create a navigation bar with links to sections like "About," "Speakers," "Agenda," "Registration," and "Contact."
   c. In the "About" section, provide a brief description of the conference's theme and purpose.
   d. Design a "Speakers" section that lists at least three keynote speakers with their names, titles, and photos.
   Note: Use semantic elements to structure this section and apply suitable CSS.
3. Demonstrate a simple webpage layout containing text and an image using CSS selectors
   a. Apply CSS styles to change the font family, size, color, and line height of the text content.
   b. Use selectors to target specific headings and paragraphs.
   c. Select the image using an element selector and apply a border with a defined width and color.
   d. Implement a hover effect that slightly increases the image's size
   e. Use class selectors to adjust the text alignment and spacing within paragraphs.
   f. Apply margin and padding to create a balanced layout.
   g. Apply a background color to the entire layout using an element selector.

4. Develop an HTML table to display the weekly class timetable, spanning Monday through Friday.
   a. Populate the table with precise class information, including course codes
   b. Utilize colspan or rowspan to merge cells horizontally or vertically, creating space for breaks or gaps in the schedule.
   c. Implement distinct background colors for cells to differentiate between different subjects and breaks.
5. Develop a student registration form
   a. Include fields for the student's name, USN, email id, address, radio button for gender and a checkbox for subject preferences (Web, Java, Python).
   b. option to upload photo and dropdown list for payment method
   c. Include a "Submit" button to process the registration

6. Build a multicolumn layout using float element in HTML. Create an HTML structure that includes a main content area and a sidebar. Use CSS floats to position the main content on the left and the sidebar on the right. Populate the main content area with blog articles, including headings, images, and text. Populate the sidebar with widgets such as recent posts, categories, and social media links.
7. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.
8. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays "TEXT SHRINKING" in BLUE color. Then the font size decreases to 5pt.
9. Develop a webpage containing 3 overlapping images using HTML, CSS and JS. Further when the mouse is over any image, it should be on the top and fully displayed.
10. Demonstrate a simple portfolio webpage using bootstrap to showcase your skills and projects

Course Outcomes: Upon successful completion of this course, student will be able to

CO35.1	Adapt HTML and CSS syntax and semantics to build web pages
CO35.2	Construct and visually format tables and forms using HTML and CSS
CO35.3	Build multicolumn layouts for build web pages
CO35.4	Develop client side scripting using javascript and analyze the Document Object Model
CO35.5	Develop responsive designs for web pages using Bootstrap

Textbooks:

1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 4th Edition, Pearson Education India, 2016.
2. Jake Spurlock, "Bootstrap: responsive web development", O'Reilly Media, Inc., 2013.

Reference Books:

1. Jon Duckett, "HTML and CSS: Design and Build Websites", 1st Edition, Wiley, 2011.
2. David DuRocher, "HTML and CSS Quickstart Guide", Clydebank Media LLC, 2021.
3. Elizabeth Robson and Eric Freeman, "Head First HTML and CSS", Second Edition, O'Reilly, 2012.
4. Istvan Novak, "Unraveling Bootstrap 3.3", Kindle Edition, 2014.

E-Books / Web References

1. http://www.pearsonglobaleditions.com/connolly
2. https://www.w3schools.com/html/html5_intro.asp
3. https://www.w3schools.com/css/
4. https://www.w3schools.com/jS/default.asp
5. https://www.w3schools.com/bootstrap/default.asp
6. https://getbootstrap.com/

MOOCs

1. NPTEL: https://nptel.ac.in/courses/106105084
2. Udemy: https://www.udemy.com/topic/web-app-development/

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO35.1	2	2	2	-	3	-	-	-	-	-	-	2	2	1
CO35.2	2	2	2	-	3	-	-	-	-	-	-	2	2	1
CO35.3	2	2	2	-	3	-	-	-	-	-	-	2	2	1
CO35.4	2	2	2	-	3	-	-	-	-	-	-	2	2	1
CO35.5	2	2	2	-	3	-	-	-	-	-	-	2	2	1
Average	**2**	**2**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	1

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20

marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

SUBJECT: UNIX AND SHELL PROGRAMMING (Integrated)

Subject Code	**22CML36**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:0:2**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**03**
No. of Credits: 3			

Course Learning Objectives

The course will enable students to

CLO1	Understand the features, architecture of UNIX and its commands.
CLO2	Discuss different UNIX files, attributes and permissions.
CLO3	Discuss filter programs and regular expressions.
CLO4	Understand essential facets of shell programming in order solve the shell script Problems.

CONTENTS	# of Hours / RBT Levels
MODULE 1 UNIX Architecture and Command Usage: Unix Architecture, Features of UNIX, Internal and External Commands General-Purpose Utilities: cal, date, echo, printf, bc, passwd, who, uname, tty, stty. The File System: The Parent-Child Relationship, the HOME variable, pwd, cd, mkdir, rmdir, Absolute Pathnames, Relative Pathnames **Text book: T1** **Chapters: 2.1, 2.2, 2.5, 3.1 to 3.5, 3.9 to 3.13, 4.1 to 4.10**	**08** **L2**
MODULE 2 Handling Ordinary Files: cat, cp, rm, mv, more, file, wc, cmp, comm, diff, Basic File Attributes: ls –l, file ownership, file permissions, chmod, directory permissions, changing file ownership More File Attributes: File Systems and Inodes, Hard Links, Symbolic Links and ln, umask Modification and Access Times **Text book: T1** **Chapters: 5.1 to 5.5, 5.10 to 5.12, 6.1 to 6.7, 11.1 to 11.6**	**08** **L3**
MODULE 3 Simple Filters: The sample database, head, tail, cut, paste, sort, uniq, tr Filters using Regular Expression: grep. egrep **Text book: T1** **Chapters: 12.1, 12.3 to 12.9, 13.1 to 13.3**	**08** **L3**
MODULE 4 Essential Shell Programming Part I: Shell Scripts, read, Using command line arguments, exit and exit status of command, the logical operators && and || - conditional execution **Text book: T1** **Chapters: 14.1 to 14.5**	**08** **L2**
MODULE 5 Essential Shell Programming Par II : the if conditional, using test and [ ] to evaluate expressions, the case conditional, expr, $0, while, for, set and shift **Text book: T1** **Chapters: 14.6 to 14.13**	**08** **L3**

Laboratory Component

List of Experiments

1. Working on general purpose commands and General-Purpose Utilities
2. Working on general purpose commands and General-Purpose Utilities (contd)
3. Working on Ordinary files and Attributes
4. Working on Ordinary files and Attributes (contd)
5. Working on Filter programs and filters using regular expression
6. Working on Filter programs and filters using regular expression
7. a) Write a shell script which displays a list of all the files in the current directory to which you have read, write and execute permissions.

   b) Write a shell script which will accept a filename and starting and ending line numbers and displays these lines from given file.

   c) Write a shell script which is expected to accept two filenames as its arguments. Check the number of arguments and display the contents of the argument files if the arguments is two, otherwise display an error message an exit.
8. a) Write a shell program that will do the following tasks in order:

   i) clear the screen ii) print the current directory iii) display current login users
   iv) list of users v) list of processes vi) list of files
   vii) today's date viii) Quit to UNIX

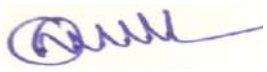
   b) Write a shell script that reads data from a text file and appends the lines starting with an alphabet from 'a' through 'k' to a file named "ak" and the lines from 'l' through 'r' to a file named "lr" and the lines from 's' through 'z' to a file named "sz". Then display the number of lines in each of the files ak, lr and sz individually.

   c) Write a shell script which will receive login name during execution, obtain information about it from /etc/passwd and display this information on screen in easily understandable format.
9. a) A shell script receives even number of filenames as arguments. Suppose four files are supplied as arguments then the first file should get copied into second, third file into fourth and so on. If odd number of filenames is supplied then no copying should take place and an error message should be displayed.

   b) Write a shell script which will receive any number of filenames as arguments. The shell script should check whether every argument supplied is a file or a directory. If it a directory it should be appropriately reported. If it is a filename then name of the file as well as the number of lines present in it should be reported.
10. a) Write a shell script which expects two parameters, a file name and a number. The script deletes the line with given number from given file.

    b) Write a shell script which accepts any number of arguments and prints them in reverse order. Ex : If file name is test then $ sh test A B C should produce C B A.

    c) Write script average which computes the average value (rounded to an integer as computed with expr) of the numeric values given in the parameters and writes the result to standard output. For example average 4 13 112 7 outputs 34.



Course Outcomes:

Upon successful completion of this course, student will be able to

CO36.1	Explain the fundamental concepts of UNIX Operating system along with the working of various commands
CO36.2	Illustrate various filters to solve variety of applications
CO36.3	Write Regular expressions for pattern matching
CO36.4	Write shell scripts

Textbooks:

1. UNIX – Concepts and Applications, Sumitabha Das, 4th Edition, McGraw Hill, 2017.

Reference Books:

1. UNIX and SHELL Programming, Behrouz A Forouzan and Richard F Gilberg, India Edition, Cengage Learning, Third Reprint 2008
2. UNIX – The Complete Reference, Kenneth Rosen et al, 2nd Edition, Tata McGraw Hill Fourth Reprint 2008

E-Books / Web References

1. http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html
2. https://www.tutorialspoint.com/unix/index.htm

MOOCs

1.https://nptel.ac.in/courses/117106113
2. https://onlinecourses.swayam2.ac.in/aic20_sp05/preview
3. http://elearning.vtu.ac.in/econtent/courses/video/CSE/CS36.htm

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO36.1	3	3	2	-	3	-	-	-	-	-	-	2	1	-
CO36.2	3	3	2	-	3	-	-	-	-	-	-	2	1	-
CO36.3	3	3	2	-	3	-	-	-	-	-	-	2	1	-
CO36.4	3	3	2	-	3	-	-	-	-	-	-	2	1	-
Average	**3**	**3**	**2**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**1**	**-**

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

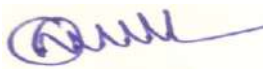
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**



SEMESTER – IV

Course: Probability and Linear Algebra

(Common for CSE/ISE/AI&DS/AIML/CS(AIML))

Course Code	**22MAT41A**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:2:0**	**SEE Marks**	**50**
No. of Credits	**3**	**Examination Hours**	**03**

Course Objectives: To enable students to apply the knowledge of Mathematics in fields of computer science and allied branches by making them to learn:

CLO1	Probability and Random Variables
CLO2	System of linear equations
CLO3	Vector spaces, linear transformations
CLO4	Eigenvalues, Eigenvectors, diagonalization and Singular value decomposition

Content	**No. of Hours/ RBT levels**
Module 1 Random Variable, Binomial, Poisson, Exponential and Normal distributions. Joint distributions (both discrete and continuous), Expectation and Covariance. Central limit theorem and law of large numbers.	**08 Hours L2, L3**
Module 2 System of linear equations, row reduction and echelon form, vector equations, The matrix equation AX = b. Linear independence and introduction to linear transformations. Matrix of linear transformation, invertible matrix, inverse of a matrix by Gauss Jordan method.	**08 Hours L2, L3**
Module 3 Vector space, subspaces, linearly independent sets, Bases. Coordinate systems, the dimensions of a vector space, Rank, Change of basis. Eigen vectors and Eigen values, diagonalization, Eigen vectors and linear transformations.	**08 Hours L2, L3**
Module 4 Inner products; inner product spaces; orthogonal sets and projections; Gram-Schmidt process; QR-factorization.	**08 Hours L2, L3**
Module 5 Least square solutions and fittings, diagonalization of symmetric matrices, quadratic forms, constrained optimization; Singular value decomposition.	**08 Hours L2, L3**

COURSE OUTCOMES:

Upon completion of this course, student will be able to:

CO41.1	Solve problems associated with random variables using probability distributions
CO41.2	Solve systems of linear equations.
CO41.3	Work within vector spaces.
CO41.4	Use computational techniques for the study of Eigenvalues, Eigenvectors, and diagonalization

Textbooks:

1. T Veerarajan, Probability, Statistics and Random Processes for Engineers, Tata McGraw Hill, 3rd Edition, 2008
2. David C Lay, Linear Algebra and its applications, Pearson, 4th Edition, 2012.

Reference books:

1. Richard H Williams, Probability, Statistics and Random Processes for Engineers, Cengage Learning, 1st Edition, 2003
2. Gilbert Strang, Linear Algebra and its Applications, Cengage Learning, 4th Edition, 2006
3. K. Hoffman and R. Kunze, Linear Algebra, Prentice Hall, 2nd Edition, 2004.

Scheme of Examination:

Semester End Examination (SEE):
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of three sub questions) from each module carrying 20 marks each. Students are required to answer any **five full questions** choosing at least **one full question from each module.**

Continuous Internal Evaluation (CIE):
Three Tests are to be conducted for 40 marks each. Average of Marks scored in all three tests is added to test component. CIE is executed by way of quizzes / Alternate Assessment Tools (AATs), and three tests. **Some possible AATs:** seminar/assignments/ mini-projects/ concept videos/ partial reproduction of research work/ group activity/ any other.

Typical Evaluation pattern for regular courses is shown in Table 2.

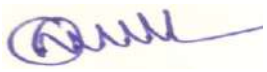
Table2 : Distribution of weightage for CIE & SEE of Regular courses

Component		Marks	Total Marks
CIE	CIE Test-1	**40**	**50**
	CIE Test-2	**40**	
	CIE Test-3	**40**	
	Assignments	**10**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

CO/PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO41.1	3	2	1									3				
CO41.2	3	2	1									3				
Average	3	2	1									3				

Low-1: Medium-2: High-3



SUBJECT: DATABASE MANAGEMENT SYSTEMS
(Integrated)

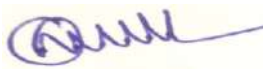
Subject Code	**22CML42**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:2**	**SEE Marks**	**50**
Total Hours	**50**	**Examination Hours**	**3**
No. of Credits: 4			

Course Learning Objectives:

The course will enable students to

CLO1	Outline a strong foundation in database concepts, technology, and practice.
CLO2	Identify a strong foundation on normalization techniques to design a database
CLO3	Demonstrate the use of concurrency and transactions in database
CLO4	Design and build database applications for real world problems

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction:** Introduction, An example, Characteristics of Database approach, Advantages of using DBMS approach, Data models, schemas and instances, Three-schema architecture and data independence **Entity-Relationship Model:** An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Sample ER Diagrams. **Text Book1: Chapter 1.1-1.3, 1.6 , 2.1,2.2, 3.2-3.6**	**10** **L3**
MODULE 2 **Relational Model and Relational Algebra:** Relational Model Concepts, Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations: JOIN and DIVISION, Examples of Queries in Relational Algebra. **Mapping Conceptual Design into Logical Design:** Relational Database design using ER to Relational Mapping. **Text Book1: Chapter 5.1, 8.1-8.3, 8.5, 9.1**	**10** **L3**
MODULE 3 **SQL:** SQL Data Definition and Data Types, Specifying basic constraints in SQL, Retrieval queries in SQL, Insert, Delete, Update statements in SQL. **SQL Advanced Queries:** More complex SQL Queries, Specifying Constraints as Assertions and Action Triggers, Views in SQL, Schema change statements in SQL. **Database Application Development:** PL/SQL, syntax, examples, create & drop procedure; If and Loops in Procedure; Introduction to Cursor, Cursor within *for* loops, Table within cursors. **Text Book1: Chapter 6.1-6.4, 7.1-7.4**	**10** **L3**
MODULE 4 **Database Design:** Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, Second, Third and Boyce Codd Normal Forms. Hashing in DBMS: Static and Dynamic Hashing Techniques. Indexing in DBMS: bitmap indexing	**10** **L3**



NoSQL Databases: What is it and Why you need it?, Basics of Graph Databases, Document-Based NOSQL Systems and MongoDB. **Text Book1: Chapter 14.1-14.5, 24.1, 24.3**	
MODULE 5	
Transactions Management: Introduction to Transaction Processing, Transaction states, Desirable properties of Transactions, Characterizing Schedules based on recoverability, Characterizing Schedules based on Serializability. **Concurrency Control and Recovery System:** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Shadow paging, ARIES recovery algorithm. **Text Book1: Chapter 20.1, 20.2.1, 20.3-20.5, 21.1, 21.2, 22.4, 22.5**	**10** **L2**

Laboratory

ComponentNote:

- Design, develop, and implement the specified queries for the following problems using Oracle,MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.

List of Experiments

1. Consider the following schema for a Library

   Database:BOOK (Book_id, Title,

   Publisher_Name, Pub_Year)

   BOOK_AUTHORS (Book_id, Author_Name)

   PUBLISHER (Name, Address, Phone)

   BOOK_COPIES (Book_id, Branch_id, No-

   of_Copies)

   BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out,

   Due_Date)LIBRARY_PROGRAMME (Branch_id, Branch_Name,

   Address)

   CARD (Card_No)

   Write SQL queries to

   a. Retrieve details of all books in the library – id, title, name of publisher, authors, number ofcopies in each Programme, etc.
   b. Get the particulars of borrowers who have borrowed more than 2 books, in the year 2020.
   c. Delete a book in BOOK table. Update the contents of other tables to reflect this datamanipulation operation.
   d. Display the total number of books published by each Publisher.
   e. Create a view of all books and its number of copies that are currently available in the Library.

2.Consider the schema for Company Database:

   EMPLOYEE (SSN, FName, LName, Address, Gender, Salary,

   DNo)DEPARTMENT (DNo, DName, MgrSSN,

   MgrStartDate) DLOCATION (DNo,DLoc)

   PROJECT (PNo, PName, PLocation,

   DNo)WORKS_ON (SSN, PNo,

   Hours)

Write SQL queries to

a. Convert employee name into uppercase whenever an employee record is inserted or updated.Trigger to fire before the insert or update.
b. Make a list of all project numbers for projects that involve an employee whose last name is‘Scott’, either as a worker or as a manager of the department that controls the project.
c. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percentraise.
d. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as themaximum salary, the minimum salary, and the average salary in this department.
e. Retrieve the name of each employee who works on all the projects controlled by departmentnumber 5.

3. The commercial bank wants keep track of the customer’s account information. Each customer may have any number of accounts and account can be shared by any number of customers. The system willkeep track of the date of last transaction.

   We store the following details.

   a) Account: unique account-number, type and balance

   b) Customer: unique customer-id, name and several addresses composed of street, city and statePerform the following operations on the database:

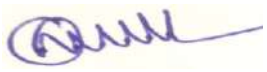
   a. Create necessary tables and insert few tuples to all the relations.
   b. Add 5% interest to the customer who have less than 10000 balance.
   c. List joint accounts involving more than three customers.
   d. Find the total interest credited to each customer for a particular year.
   e. Find the customer who has not done any transaction.

4. A database is to be designed for a college to monitor students' progress throughout their course of study.The students are reading for a degree (such as B.E.) within the framework of the modular system. Thecollege provides a number of Subjects (Modules), each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from, prerequisite course. Department may be CSE, ISE etc. A Subject is co- ordinated by a module leader who shares teaching duties with one or more teachers. A Teacher may teach (and be a module leader for) more than one Subject. Students are free to choose any subject they wish. The database also contains some informationabout students including their Serial numbers, names, addresses, their past performance (i.e. subjects taken and Subject Examination Marks).

   For this case study,

   a. Analyze the data required, create the tables and insert the values.
   b. Retrieve the Teacher names who are not Module leaders.
   c. Display the department which offers the subject “Database Management System”.
   d. Display the number of Subjects taught by each Teacher.
   e. Categorize students based on the following criterion: If Subject Examination Marks = 70 to 100 then CAT = ‘Outstanding’ If FinalIA = 40 to 69 then CAT = ‘Average’ If FinalIA< 39 then CAT = ‘Weak’.

5. A college consists of number of employees working in different departments. In this context, create two tables’ employee and department.



Employee consists of columns Empno, Empname, Basic, HRA, DA, Deductions, Gross, Net, Date-of-birth. The calculation of HRA, DA are as per the rules of the college. Initially only Empno, Empname,Basic have valid values. Other values are to be computed and updated later. Department contains Deptno, Deptname, and Department Location columns. Deptno is the primary key in department tableand referential integrity constraint exists between employee and department tables.

Perform the following operations on the database:

a. Create the tables employee and department with proper constraints.

b. Add constraint that basic should not be less than 5000.

c. Calculate HRA, DA, gross and net by using PL/SQL program.

BASIC	DA	HRA
15000	12%	8%
12000	10%	6%
9000	7%	4%
OTHERS	5%	200/-

d. Write PL/SQL program that whenever salary is updated and its value becomes less than 5000 atrigger has to be raised preventing the operation.

Course Outcomes

Upon successful completion of this course, student will be able to

CO42.1	Construct ER models to represent simple database applications.
CO42.2	Develop Relational Algebraic expressions for complex Relational Algebra operations.
CO42.3	Develop SQL/PL/SQL programs for queries using Relational Model Concepts.
CO42.4	Demonstrate relational database model for an application by normalizing the database schema.
CO42.5	Illustrate the use of concurrency control and transactions in database.

Text books:

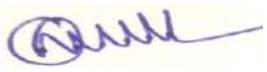
1. Elmasri, Ramez, and Sham Navathe. Fundamentals of database systems. Vol. 7. Pearson, 2014.
2. Tiwari, Shashank. Professional nosql. John Wiley & Sons, 2011.

Reference Books:

1. Raghurama Krishnan, Johannes Gehrke , Database Management Systems, 3rd edition, Tata McGrawHill, New Delhi, India.
2. Silberschatz, Korth and Sudharshan: Database System Concepts, 6th Edition, Mc-GrawHill, 2010.
3. C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, PearsonEducation, 2006.
4. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

E-Books / Web References

1. http://www.mim.ac.mw/books/Elmasri-Navathe-Fundamentals-of-Database-Systems-5th-Editi.pdf
2. Procedures in SQL | SQL Stored Procedures | Edureka



3. Hashing in DBMS | 2 Main Types of Hashing Techniques in DBMS (educba.com)
4. Blog Theme - Details (oracle.com)
5. Bitmap Indexing in DBMS - GeeksforGeeks
6. GridFS — MongoDB Manual

MOOCs

1. https://archive.nptel.ac.in/courses/106/105/106105175

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO42.1	3	-	-	-	-	-	-	-	-	-	-	2	-	2
CO42.2	3	3	2	-	-	-	-	-	1	1	-	2	-	2
CO42.3	3	3	1	-	2	-	-	-	1	1	-	2	-	2
CO42.4	3	3	1	-	2	-	-	-	1	1	-	2	-	2
CO42.5	1	1	-	-	-	-	-	-	1	1	-	2	-	2
Average	**3**	**3**	**1**	**-**	**2**	**-**	**-**	**-**	**1**	**1**	**-**	**2**	**-**	**2**

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**

SUBJECT: DESIGN AND ANALYSIS OF ALGORITHMS (Integrated)

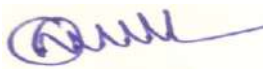
Subject Code	**22CML43**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:2**	**SEE Marks**	**50**
Total Hours	**50**	**Examination Hours**	**3**
No. of Credits: 4			

Course Learning Objectives:

The course will enable students to

CLO1	Apply mathematical concepts and notations to define a problem.
CLO2	Understand and apply algorithms design techniques
CLO3	Gain ability to solve real life problems using algorithms techniques.
CLO4	Understand the limitations of Algorithmic power.

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction:** What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Algorithm Specification, Performance Analysis: Space complexity, Time complexity Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples. **Text Book 1 : Chapters 1, 2.1-2.4** **Text Book 2 : Chapters 1.2, 1.3**	**10** **L2**
MODULE 2 **BRUTE FORCE:** Brute force string matching algorithms. **DIVIDE & CONQUER:** General method, Recurrence equation for divide and conquer, Binary Search, Merge sort, Quick sort, Strassen's matrix multiplication, Advantages and Disadvantages of divide and conquer. **Text Book 1 : Chapters 3.2, 5.1, 5.2, 5.4** **Text Book 2 : Chapters 3.1, 3.3**	**10** **L3**
MODULE 3 **GREEDY METHOD:** Introduction, General method, Knapsack Problem, Job sequencing with deadlines**,** **Minimum cost spanning trees:** Prim's Algorithm, Kruskal's Algorithm, **Single source shortest paths:** Dijkstra's Algorithm**, Optimal Tree problem:** Huffman Trees and Codes **Transform and Conquer Approach:** Heaps and Heap Sort, AVL Tree, 2-3 Tree **Text Book 1 : Chapters 6.3-6.4,9.1-9.4** **Text Book 2 : Chapters 4.3,4.5**	**10** **L3**
MODULE 4 **DYNAMIC PROGRAMMING:** Introduction, Transitive closure - Warshall's and Floyds algorithm, Knapsack problem & memory functions, Bellman Ford algorithm. **DECREASE & CONQUER:** Introduction – Decrease by constant, decrease by constant factor, variable size decrease, Breadth First search traversal, Depth First search traversal, Topological sorting using DFS and source removal method. **Text Book 1 : Chapters 3.5, 4.2, 8.2, 8.4** **Text Book 2 : Chapters 5.4,5.9**	**10** **L3**



MODULE 5 **BACKTRACKING:** N-Queens problem, Sum of subsets problem, Hamiltonian cycles **BRANCH & BOUND:** Introduction, Travelling Salesman problem, Knapsack problem, Assignment problem **LIMITATIONS OF ALGORITHM POWER:** Decision Trees for sorting and searching Approximation Algorithms for NP-Hard Problems – Traveling Salesperson Problem using Nearest-neighbor algorithms **Text Book 1 : Chapters 11.2-11.3, 12.1-12.2**	**10** **L3**

Laboratory Component

List of Experiments

1. Sort a given set of n integer elements using Quick Sort method and compute its time complexity.
2. Sort a given set of n integer elements using Merge Sort method and compute its time complexity.
3. Find Minimum Cost Spanning Tree of a given connected undirected graph using
   i) Kruskal's algorithm. ii) Prim's algorithm.
4. Write a program to find shortest path using Dijkstra's algorithm.
5. Write a program to implement All-Pairs Shortest Paths problem using Floyd's algorithm.
6. Solve the given instance of 0/1 Knapsack problem using Dynamic Programming
7. Write a program to Print all the nodes reachable from a given starting node in a digraph using BFS method.
8. Write a program to Check whether a given graph is connected or not using DFS method.
9. Solve N-Queen's problem using Back Tracking.
10. Develop Back Tracking solution for Sum of Subset Problem.

Course Outcomes:

Upon successful completion of this course, student will be able to

CO43.1	Explain the basic techniques of analyzing the algorithms using time & space complexity and asymptotic notations
CO43.2	Devise algorithms using brute force and Divide and Conquer techniques for a given problem.
CO43.3	Demonstrate Graph Algorithms using greedy method, Transform and Conquer Approach to model Engineering Problems.
CO43.4	Employ Dynamic Programming and Decrease & Conquer strategies to solve a given problem
CO43.5	Use Back Tracking, Branch and Bound design techniques for solving Computationally hard problems.

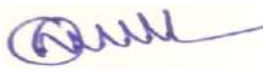
Textbooks:

1. Introduction to The Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Pearson, Tenth Impression 2020.
2. Computer Algorithms/C++, Ellis Horowitz, Sartaj Sahni & Sanguthevar Rajasekaran, University Press, 2nd Edition, Reprint 2017.

Reference Books:

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2022.

E-Books / Web References



1. http://www.facweb.iitkgp.ac.in/~sourav/daa.html
2. https://freevideolectures.com/course/2281/design-and-analysis-of-algorithms

MOOCs

1. https://nptel.ac.in/courses/106101060/
2. https://www.coursera.org/specializations/algorithms

Mapping of CO-PO

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO43.1	3	3	3	-	3	-	-	-	-	-	-	2	2	3
CO43.2	3	3	3	-	3	-	-	-	-	-	-	2	2	3
CO43.3	3	3	3	-	3	-	-	-	-	-	-	2	2	3
CO43.4	3	3	3	-	3	-	-	-	-	-	-	2	2	3
CO43.5	3	3	3	-	3	-	-	-	-	-	-	2	2	3
Average	**3**	**3**	**3**	**-**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**3**

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

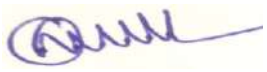
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**



SUBJECT: SOFTWARE ENGINEERING

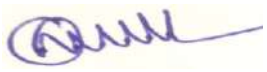
Subject Code	**22CML44**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**3:0:0**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**3**
No. of Credits: 3			

Course Learning Objectives:

The course will enable students to

CLO1	Outline software engineering principles and activities involved in building large software programs.
CLO2	Introduce Software design using UML tools and various system models.
CLO3	Introduce Agile Software development and Agile methods – SCRUM.
CLO4	Introduce Software Testing and Software Evolution processes in detail.
CLO5	Explain project planning, Cost estimation techniques and Software quality, standards and metrics.

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction:** Professional Software Development - Software Engineering, Case Studies. **Software Processes:** Software process models - Waterfall Model, Incremental Model, and Spiral Model, Software Process Activities. **Requirement Engineering:** Requirements Specification, Requirement elicitation and analysis, Requirements Validation, Requirements change, Functional Specification, and Non-Functional Requirements, The software requirements document. **Case Study:** Create a SRS document for the following software projects. 1. Online Ticket Reservation System 2. Stock Maintenance 3. Student's Marks Analyzing System 4. Stock Maintenance **Text book:1 Chapters: 1.1-1.3,2.1-2.2,4.1-4.6**	**08** **L2**
MODULE 2 **System Models:** Context models. Interaction models. Structural models. Behavioral models. Model-driven engineering. **Software Design and Planning:** Object-oriented design concepts using UML tool: Star UML application, Design patterns, Implementation issues, Open-Source Development. **Text book:1 Chapters:5.1-5.5,7.1-7.4**	**08** **L2**
MODULE 3 **Agile Software Development:** Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods. SCRUM Methodology, SCRUM. **Text book:1 & 3 Chapters:3.1-3.4**	**08** **L2**
MODULE 4 **Software testing strategies:** A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional software, Test Strategies for Object-Oriented Software, Test Strategies for web apps, Validation Testing, and System Testing.	**08** **L2**



Testing Conventional Applications: Software Testing Fundamentals, Internal and External views of Testing, White-Box Testing, Basic path Testing, Control structure Testing, and Black-box Testing. **Case study:** 1. Design the test case for finding the roots of the quadratic equation. 2. Design the test case for the e-commerce application. **Text book:2 Chapters: 17.1-17.7,18.1-18.6**	
MODULE 5 **Project planning:** Software pricing, Plan-driven development, Project scheduling, Agile planning, and Estimation techniques. **Quality management:** Software quality, Software standards, Reviews and inspections, Software measurement, and metrics. **Text book:1 Chapters:23.1-23.5,24.1-24.3 & 24.5**	**08** **L2**

Course Outcomes:

Upon successful completion of this course, student will be able to

CO44.1	Understand the fundamentals of Software Engineering, Software process models, and Requirements Engineering.
CO44.2	Summarize the different types of System Models, Software Design and Planning by using UML tool.
CO44.3	Outline Agile Software Development and Agile Methods – SCRUM.
CO44.4	Describe Software testing methods and Conventional Applications.
CO44.5	Discuss the project planning process, Cost estimation models, Software Quality standards, and metrics.

Textbooks:

1. Software Engineering, Ian Sommerville, 10th Edition, Pearson Education, 2016.
2. Software Engineering: A Practitioner's Approach, Roger S Pressman, 7th Edition, Tata McGraw-Hill, 2014.
3. The SCRUM Primer, Ver 2.0, http://www.goodagile.com/scrumprimer/scrumprimer20.pdf

Reference Books:

1. Pankaj Jalote, An integrated approach to software engineering, Springer US, 3rd Edition, 2005.
2. Michael Blaha, James Rumbaugh, Object Oriented Modelling and Design with UM, Pearson Education, 2nd Edition, 2005.
3. Rajib Mall, Fundamentals of Software Engineering, 4th Edition, PHI Learning Private Limited,2014.
4. Tom Pender, UML Bible, Wiley Publishing, 2003.

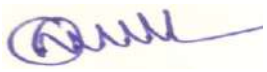
E-Books / Web References

1. https://www.softwaretestingmaterial.com/category/agile/
2. https://www.atlassian.com/agile/kanban

MOOCs

1. https://www.coursera.org/learn/introduction-to-software-engineeringE-learning: www.vtu.ac.in

Mapping of CO-PO:



CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO44.1	3	-	-	-	-	1	1	1	-	3	-	2	2	2
CO44.2	3	-	3	-	-	1	1	1	-	3	-	2	2	2
CO44.3	3	-	3	-	-	1	1	1	-	3	-	2	2	2
CO44.4	3	3	3	3	-	3	-	-	3	3	3	3	2	2
CO44.5	3	1	-	-	-	1	1	1	-	3	2	2	2	2
Average	**3**	**2**	**3**	**3**	**-**	**2**	**1**	**1**	**3**	**3**	**3**	**3**	**2**	**2**

Low-1: Medium-2: High-3

Scheme of Evaluation:

Semester End Examination (SEE):

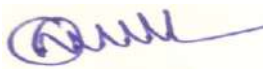
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three tests are to be conducted for 40 marks each. Average of all three tests marks is added to the test component. In order to encourage innovative methods while delivering a course, the faculty members have been encouraged to use the Alternative Assessment Tool (AAT). The AAT enhances the autonomy (freedom and flexibility) of individual faculty and enables them to create innovative pedagogical practices. Possible AATs are – seminar / assignments / term paper / open ended experiments / mini-projects/group activity or any other.

Table: Distribution of weightage for CIE & SEE

	Component	Marks	Total Marks
CIE	CIE Test-1	**40**	**50**
	CIE Test-2	**40**	
	CIE Test-3	**40**	
	AAT	**5**	
	AAT	**5**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**



SUBJECT: OBJECT ORIENTED PROGRAMMING WITH JAVA (Integrated)

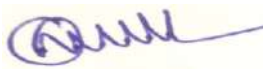
Subject Code	**22CML45**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:0:2**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**3**
No. of Credits: 3			

Course Learning Objectives:

The course will enable students to:

CL01	Set up Java JDK environment to create, debug, and run simple Java programs.
CLO2	Learn object-oriented concepts using programming examples.
CLO3	Create multi-threaded programs and event-handling mechanisms.
CLO4	Learn string handling methods using programming examples.
CLO5	Create Interactive programs.

CONTENTS	# of Hours / RBT Levels
MODULE 1 **Introduction to JAVA:** Java's magic, Features, the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs' Data types, variables, Operators, and Control Statements. **Classes & Objects:** Classes fundamentals; Declaring objects; Constructors, this keyword, static keyword, and compile time polymorphism. **Textbook: 1** **Chapters: 1, 2, 3, 4, 5, 6, 7**	**8** **L3**
MODULE 2 **Inheritance:** inheritance basics, uses of super keyword, run time polymorphism, Final and Abstract keywords. **Exception handling:** Fundamentals, Exception Types, Uncaught Exceptions, Exception Handlers, Java's Built-in Exceptions, Creating user-defined Exception Subclasses, Chained Exceptions. **Textbook: 1** **Chapters: 8 and 10**	**8** **L3**
MODULE 3 **Arrays**: Introduction, Declaration, and Initialization, Alternative Array Declaration Syntax, One-Dimensional, Multi-Dimensional Arrays and Array of Objects. **String:** The String Constructors, String Length, Special String Operations. String Literals, String Concatenation, String Concatenation with Other Data Types. String Conversion toString(). Character Extraction methods, String Comparison methods, Searching Strings, and Modifying a String. Data Conversion using valueOf(), Changing the Case of Characters Within a String. Jumping Strings.	**8** **L3**



StringBuffer, StringBuffer Constructors, length() and capacity(). ensureCapacity(), setlength(), and setCharAl(), getChars(), append( ), insert(), reverse(), delete() and deleteCharAt(), replace( ). **Textbook: 1** **Chapters: 3 and 17**	
MODULE 4 **Packages and Interfaces:** Packages, Access Protection, Import Packages, Interfaces. **Multi-Threaded Programming**: What are threads?; Ways of Creating a Thread, Main Thread, Thread States, Synchronization, isAlive(), Join(), Priority(); Inter-Thread Communication with Producer-Consumer Problem. **Textbook: 1** **Chapters: 9 and 11**	**8** **L3**
MODULE 5 **Abstract Window Tool Kit Basics:** creating Frame, adding components like Buttons, labels, Textfield, Layouts, etc. and built-in functions. **Event Handling:** Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces, and Adapter Classes. Keyboard and Mouse Events. **Textbook: 1** **Chapters: 25, 26, and 10**	**8** **L3**

Laboratory Component

Lab Experiments:

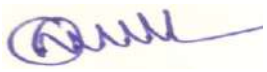
1. Develop a pizza ordering application in java, such that It creates a Pizza object to the specifications that the user desires. It walks the user through ordering, giving the user choices, which the application then uses to decide how to make the pizza and how much the cost of the pizza will be. The user will also receive a 20% discount if his/her Token number is even and divisible by 4.

2. Implement a Java application for Inventory Management System.

3. Develop a Java program that calculates and displays the area of geometric shapes using Inheritance and exception handling. The program must define a base class Shape with derived classes Circle and Rectangle. A custom exception, InvalidDimensionException, is thrown when negative dimensions are encountered.

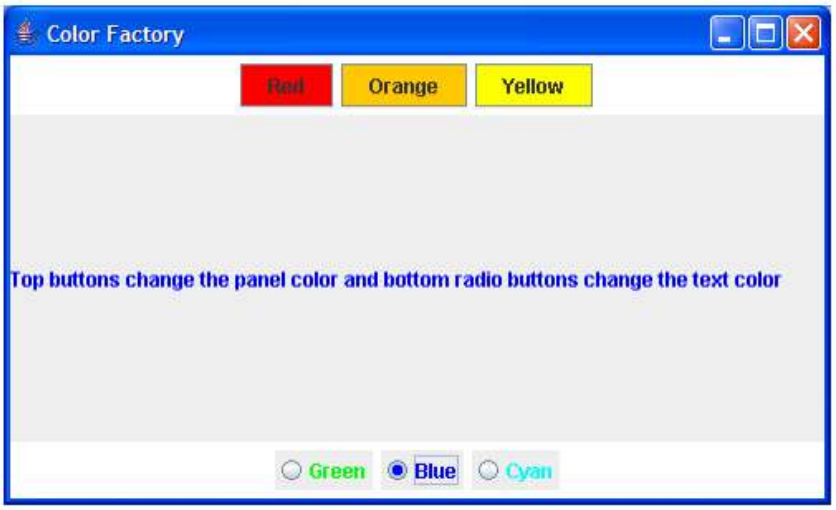
4. Extend the java application designed in Q2 to illustrate the concept of Inheritance and Exception.

5. Develop a Java application to perform a Word Game (simplified version of Hangman Game): which randomly selects a word from an array, and the user must guess the word letter by letter. The user has a limited number of attempts.
6. The HR department of XYZ Company wants to use a simple employee management system to keep track of their employees. The system should allow them to perform the following tasks:
    a. Add New Employees: HR can add new employees to the system by providing the employee's name, employee ID, and salary.
    b. View Employee List: HR can view the list of all employees currently in the system. The list includes each employee's ID, name, and salary.
    c. Search for Employees: HR can search for specific employees by entering their name. The system will display the details of the employee if found.
7. Implement a document printing center application using java, where various types of documents, such as text documents and image files, need to be printed. Each document type requires a different printing process. To efficiently handle the printing tasks concurrently use packages, interfaces, multi-threading.



8. Extend the java application designed in Q3 to illustrate the concept of packages, interfaces, and multi-threading.

9. Write a Java application to perform the following GUI task.



10. Develop a simple music player application using Java abstract window tool kit and Event handling mechanisms.

Course Outcomes: Upon successful completion of this course, students will be able to

CO45.1	Illustrate the fundamental concepts of Object-Oriented Programming using Java
CO45.2	Develop Java applications using Inheritance and Exception handling
CO45.3	Utilize Arrays and String handling functions to write the Java application
CO45.4	Apply Interfaces and multi-threading concepts to create parallel programming
CO45.5	Make use of Event-handling mechanisms to create interactive Java applications

Textbooks:

1. Herbert Schildt, Java The Complete Reference, 11th Edition, Tata McGraw Hill, 2019.

Reference Books:

1. Mahesh Bhave and Sunil Patekar, “Programming with Java", First Edition, Pearson Education,200B, ISBN:9788 I 3 I 720806
2. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, “Object-oriented Programming with java”, Tata McGraw Hill education private limited.
3. Cay S Horstmann and Cary Gornell, “CORE JAVA volume I-Fundamentals”, Pearson
4. James W. Cooper, “Java TM Design Patterns – A Tutorial”, Addison-Wesley Publishers.

E-Books / Web References

1. https://greenteapress.com/thinkjava6/thinkjava.pdf
2. https://books.goalkicker.com/JavaBook/

MOOCs

1. https://nptel.ac.in/courses/106105191
2. https://www.mooc-list.com/tags/java-programming
3. https://java-programming.mooc.fi/

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO45.1	3	3	3	-	3	-	-	-	1	1	-	2	1	-
CO45.2	3	3	3	-	3	-	-	-	1	1	-	2	1	-
CO45.3	3	3	3	-	3	-	-	-	1	1	-	2	1	-
CO45.4	3	3	3	-	3	-	-	-	1	1	-	2	1	-
CO45.5	1	3	3	-	3	-	-	-	1	1	-	2	1	-
Average	**3**	**3**	**3**	-	**3**	-	-	-	**1**	**1**	-	**2**	**1**	-

Low-1: Medium-2: High-3

Scheme of Evaluation: (Integrated courses)

Semester End Examination (SEE):

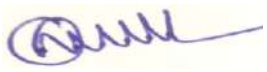
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):

Three Tests are to be conducted for 40 marks each. The average of the three tests are taken for computation of CIE on a scale of 30, the CIE would also include laboratory evaluation for 20 marks. The laboratory marks of 20 would comprise of 10 marks for regular laboratory assessment to include lab record and observation. 10 marks would be exclusive for laboratory internal assessment test to be conducted at the end of the semester.

Table: Distribution of weightage for CIE & SEE of Integrated courses

	Component	Marks	Total Marks
CIE	CIE Test-1	**30**	**50**
	CIE Test-2	**30**	
	CIE Test-3	**30**	
	Lab	**20**	
SEE	Semester End Examination	**50**	**50**
Grand Total			**100**



SUBJECT: ADVANCED PYTHON

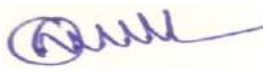
Subject Code	**22CML46**	**CIE Marks**	**50**
Hours/Week (L: T: P)	**2:2:0**	**SEE Marks**	**50**
Total Hours	**40**	**Examination Hours**	**3**
No. of Credits: 3			

Prerequisites (if any): Programming with Python.

Course Learning Objectives:

Sl. No	Course Learning Objectives (CLO)
CLO1	To build a strong foundation to understand advanced python packages for data science.
CLO2	To understand daily tasks that data scientists tackle.
CLO3	To gain skills needed to implement machine learning, AI, and predictive analytics algorithms.
CLO4	To understand the access to a wide variety of data analysis and data science libraries.
CLO5	To meet industry demand for experts with Python skills.

Module 1	Hours / RBT Levels
NumPy Basics: Arrays and Vectorized Computation **Basics of Numpy Arrays:** NumPy Array Attributes **The NumPy and array:** A Multidimensional Array Object, Creating nd arrays, Data Typesfor nd arrays, Arithmetic with NumPy Arrays, Basic Indexing and Slicing, Boolean Indexing,Fancy Indexing, Transposing Arrays and Swapping Axes, Reshaping of Arrays,Aggregations, Universal Functions: Fast Element-Wise Array Functions, Array-OrientedProgramming with Arrays, Expressing Conditional Logic as Array Operations,Mathematical and Statistical Methods, Methods for Boolean Arrays, Sorting, Unique andOther Set Logic, File Input and Output with Arrays, Linear Algebra, Pseudorandom Number Generation, Example: Random Walks.	**08 L3**
Module 2	
Pandas: Installing and using Pandas, Introducing Pandas Objects, Operating on data in pandas. **Introduction to pandas Data Structures:** Series, DataFrame, Index Objects **Essential Functionality:** Reindexing, Dropping Entries from an Axis, Indexing, Selection, and Filtering, Integer Indexes, Arithmetic and Data Alignment, Function Application and Mapping, Sorting and Ranking, Axis Indexes with Duplicate Labels. **Combining Datasets:** Concat, Append, Merge and Join. Working with Time Series.	**08 L3**
Module 3	
Plotting and Visualization: A Brief matplotlib API Primer: Figures and Subplots, Chartsusing plot(), pie chart, violin plot, scatter plot, histogram, bar chart, area plot, Quiver plot,Mesh grid, contour plot, Colors, Markers, and Line Styles, Ticks, Labels, and Legends, Annotations and Drawing on a Subplot,	**08**



Saving Plots to File, matplotlib Configuration. **Plotting with pandas and seaborn:** Three-Dimensional Plotting in Matplotlib, Python Visualization Tools for categorical Variables and Continuous Variables.	**L3**
Module 4	
Data Cleaning and Preparation: Handling Missing Data, Filtering Out Missing Data, Filling in Missing Data, Data Transformation, Removing Duplicates, Transforming Data Using a Function or Mapping, Replacing Values, Renaming Axis Indexes, Discretization and Binning, Detecting and Filtering Outliers, Computing Indicator/Dummy Variables. **Data Wrangling:** Join, Combine, and Reshape: Combining and Merging Datasets,Database-Style Data Frame Joins, Merging on Index, Concatenating Along an Axis,Combining Data with Overlap, Reshaping and Pivoting, Reshaping with Hierarchical Indexing, Pivoting "Long" to "Wide"Format, Pivoting "Wide"to "Long"Format	**08** **L3**
Module 5	
Data Preprocessing Datasets and Partitions, Data Pre-processing and Scaling: Different Preprocessing techniques, Data Integration, Outlier removal, artifact removal, Applying Data Transformations, ScalingTraining and Test Data the Same Way, Data Normalization, Data Transformation techniques.	**08** **L3**

Course Outcomes:

Upon successful completion of this course, student will be able to

CO46.1	Apply the fundamental of NumPy and Ndarrays on the real data.
CO46.2	Apply the fundamental of Pandas on the dataset for analysis.
CO46.3	Illustrate graphically data using matplotlib and seaborn libraries and results of statistical calculations.
CO46.4	Identify proficiency in the Data preparation and wrangling.
CO46.5	Apply the concepts of data preprocessing on the real world data.

Text Books:

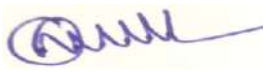
1. Python Data Science handbook, by Jake Vander Plas, O'Reilly.
2. Python for Data Analysis, by Wes McKinney, 2nd Edition, O'Reilly.
3. Bharti Motwani, Data Analytics using Python, Wiley.

Reference Books:

1.Gowrishankar S, Veena A, ―Introduction to Python Programming, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372.

2.AurelienGeron, ―Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems‖, 2nd Edition, O'Reilly Media, 2019. ISBN – 13: 978- 9352139057.

E-Books / Web References:



1. Automate The Boring Stuff With Python:
https://automatetheboringstuff.com/
2.Python3Tutorialhttps://www.tutorialspoint.com/python3/python_tutorial.pdf
3.Python for Absolute Beginners
http://indexof.es/Python/Python%203%20for%20Absolute%20Beginners.pdf

MOOCs:

1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.edx.org/course/python-basics-for-data-science
3. https://cognitiveclass.ai/courses/python-for-data-science

Mapping of CO-PO:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO46.1	3	3	3		2	2			2			2	1	2
CO46.2	3	3	3		2	3					2	2	1	2
CO46.3	3	2	3		2					3		2	1	2
CO46.4	2	3	3		3	2			3	3		2	1	2
CO46.5	3	3	3		2				2		3	3	1	2
Average	3	3	3		2	2			2	3	2	2	1	2

Low-1: Medium-2: High-3

Scheme of Evaluation:

Semester End Examination (SEE):
SEE Question paper is to be set for 100 marks and the marks scored will be proportionately reduced to 50. There will be two full questions (with a maximum of four sub questions) from each module carrying 20 marks each. Students are required to answer any five full questions choosing at least one full question from each module.

Continuous Internal Evaluation (CIE):
Three tests are to be conducted for 40 marks each. Average of all three tests marks is added to the test component. In order to encourage innovative methods while delivering a course, the faculty members have been encouraged to use the Alternative Assessment Tool (AAT). The AAT enhances the autonomy (freedom and flexibility) of individual faculty and enables them to create innovative pedagogical practices. Possible AATs are – seminar / assignments / term paper / open ended experiments / mini-projects/group activity or any other.

Table: Distribution of weightage for CIE & SEE

	Component	Marks	Total Marks
CIE	CIE Test-1	40	50
	CIE Test-2	40	
	CIE Test-3	40	
	AAT	5	
	AAT	5	
SEE	Semester End Examination	50	50
Grand Total			100

H.M. Rajashekhar Sneas
Dean Academic
Global Academy of Technology,
Rajarajeshwarinagar, Bengaluru-98

HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering (AI & ML)
Global Academy of Technology
Bangalore- 560 098.