B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Syllabus

Semester: III

Course: Fourier Transform, Fundamentals of logic and Linear Algebra

Course Code: 22MAI131 (Common to CSE, ISE, AIML)

L:T:P:J	**2:2:0:0**	**CIA**	**: 50**
Credits:	**03**	**SEA**	**: 50**
Hours:	**40**	**SEA Duration**	**: 03 Hours**

Course Learning Objectives: The students will be able to

1 Have an insight into Fourier series, Fourier transforms.

2 Develop knowledge of Fundamentals of logic and Relations, Vector Spaces & Linear Transformation arising in engineering

Module-1: Fourier Series & Fourier Transforms	**No. of hours**	**Blooms cognitive Levels**
Examples from Engineering that require Fourier series and Fourier Transforms. **Fourier series**: Periodic functions, Introduction to Fourier Series, Dirichlet's condition. Problems on Fourier series over $(-l, l)$. **Fourier Transforms:** Introduction to infinite Fourier transform, Fourier sine and cosine transform and properties, problems on infinite Fourier transform, Discrete & Fast Fourier transform. *Experiential Learning component: Finding the Fourier series and Fourier Transform of a function*	**L : 04** **T : 04**	**Apply**
Module-2: Fundamentals of logic and Relations		
Examples from Engineering that require Fundamentals of logic and Relations. **Fundamentals of logic:** Basic connectives and truth tables, logic equivalence - the laws of logic, logical implication- rules of inference **Relations:** First order linear recurrence relation, second order linear homogenous recurrence relation with constant coefficients. *Experiential Learning component: Finding the solution of recurrence relation*	**L : 04** **T : 04**	**Apply**
Module-3: Vector Spaces		
Examples from Engineering that require vector spaces Recap of system of linear homogenous and non-homogeneous equation and solution sets. Vector spaces, subspaces, linearly independent and dependent, Linear span of a set, Basis and dimension, coordinate vectors. *Experiential Learning component: Problems on linearly independent and dependent vectors, basis and dimension of a vector space.*	**L : 04** **T : 04**	**Apply**
Module-4:Linear Transformation		
Examples from Engineering that require linear transformation. Linear transformations, algebra of linear transformations, representation of transformations by matrices, Non-singular linear transformation, Inverse of a linear transformation, Range space, Null space and problems on Rank-nullity theorem. *Experiential Learning component: Problems on Inverse of a linear transformation and Rank-nullity theorem*	**L : 04** **T : 04**	**Apply**
Module-5: Inner Product Spaces		
Examples from Engineering that require Inner product spaces. Inner products Inner product spaces, Orthogonal set, orthogonal projections, orthonormal bases, Gram-Schmidt process, QR-factorization, Recap of Eigen values and Eigen vectors, problems on Singular value decomposition. *Experiential Learning component: Problems on QR-factorization and singular value decomposition*	**L : 04** **T : 04**	**Apply**

Course Outcomes: After completing the course, the students will be able to

CO 1: Apply Fourier series & transform concepts in data visualization and cryptography.

CO 2: Communicate the basic concepts of logic and their relevance for computer science engineering.

CO 3: Apply the knowledge of vector spaces for solving problems in arising in engineering field

CO 4: Apply the knowledge of linear transform for solving problems in arising in image processing

CO 5: Compute orthogonal and orthonormal bases vectors and decomposition of a symmetric matrix using standard technique.

Reference Books:

1. E. Kreyszig: “Advanced Engineering Mathematics”, John Wiley & Sons,10th Edition(Reprint), 2016.
2. B. S. Grewal: “Higher Engineering Mathematics”, Khanna Publishers, 44th Ed., 2017.
3. H. K. Dass, “ Advanced Engineering Mathematics” S. Chand publication.
4. C. Ray Wylie, Louis C.Barrett : “Advanced Engineering Mathematics", 6" Edition, 2. McGraw-Hill Book Co., New York, 1995.
5. James Stewart : “Calculus —Early Transcendentals”, Cengage Learning India Private Ltd., 2017.
6. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
7. Srimanta Pal & Subodh C Bhunia: “Engineering Mathematics”, Oxford University Press, 3"Reprint, 2016.
8. David C. Lay, Steven R. Lay and J. J. McDonald “Linear Algebra and its applications”, 3rd Edition, Pearson Education Ltd., 2017.
9. Ralph P. Grimaldi, “ Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education 2004.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111106111
2. https://youtu.be/OynpZwylau8
3. https://archive.nptel.ac.in/courses/111/106/111106051/
4. https://www.youtube.com/watch?v=zvRdbPMEMUI
5. https://www.youtube.com/watch?v=cHNmT1-qurk
6. https://www.youtube.com/watch?v=ATqV_I8DCh0

ASSESSMENT STRUCTURE:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><th></th><th>I</th><th>II</th></tr>
<tr><td rowspan="5">Conduction</td><td rowspan="5">50</td><td rowspan="5">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="3"><b>Total – 50 marks</b></td><td><b>Total – 50 marks</b></td></tr>
</table>

i) CIA: 50%

IA Test: 2 IA tests - Each of 50 Marks	Average of 2 tests – scaled down to 25 M
Assignment – Two assignments – one for 10 marks and another for 5 marks	15 Marks
Additional Assessment Tools (AAT) – Oral /Online Quizzes, Presentations, Group discussions, Case studies, Term Paper, Open ended experiments, Mini industrial/social/rural Projects, Two-minute video on latest topic, Short MOOC courses, Practical Orientation on Design thinking, creativity & Innovation, Participatory & Industry integrated learning, Practical activities, Problem solving exercises, Participation in seminars/academic events/symposia and any other activity	10 Marks
Total	**50 Marks**

ii) SEA : 50%

Theory Exam	5 questions to answer each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – III

Computer Organization and Architecture (PCC)

Credit : 3

Course Code	22AML132	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	3:0:0:0	SEA Marks	50
Total Number of Lecture Hours	40	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Explain the basic sub systems of a computer, their organization, structure and operation.
- Illustrate the concept of programs as sequences of machine instructions.
- Demonstrate different ways of communicating with I/O devices and standard I/O interfaces.
- Describe memory hierarchy and concept of virtual memory.
- Describe arithmetic and logical operations with integer and floating-point operands.
- Illustrate organization of a simple processor, pipelined processor and other computing systems

	Number of Hours	**Bloom's Level**
Module 1 : CENTRAL PROCESSING UNIT		
Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Instruction Set: CISC and RISC, Performance Measurement. **Machine Instructions and Programs:** Memory Location and Addresses, Instructions and Instruction Sequencing, Addressing Modes **Use case:** Convert High level language (C Language) to machine level language	**8**	**(CO1)** **Apply**
Module 2 : COMPUTER ARITHMETIC		
Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division **Use Case:** Data Representation and operations performed in Arithmetic Logical Unit (ALU) in computer hardware.	**8**	**(CO2)** **Apply**
Module 3 : MEMORY ORGANIZATION		
Basic Concepts, Semiconductor RAM Memories: Internal organization of memory chips, static memories, Asynchronous and synchronous DRAMs, Cache Memories – Mapping Functions.	**8**	**(CO3)** **Apply**

<table>
<tr><td><b>Use Case:</b> Helps in designing cost, space, time optimized applications
<b>Prerequisite for:</b> Operating System- Memory Management</td><td></td><td></td></tr>
<tr><td colspan="3"><b>Module 4 : INPUT - OUTPUT ORGANIZATION</b></td></tr>
<tr><td>Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits
<b>Use Case:</b> To build device drivers</td><td><b>8</b></td><td><b>(CO3)
Apply</b></td></tr>
<tr><td colspan="3"><b>Module 5 : BASIC PROCESSING UNIT</b></td></tr>
<tr><td><b>Basic Processing Unit:</b>
Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control
<b>Pipelining:</b>
Basic concepts of pipelining
<b>Use Case:</b> Optimization of Processor speed</td><td>8</td><td><b>(CO1)
Apply</b></td></tr>
</table>

Course outcomes:

The students will be able to

1. Design assembly Language program for various problems.(Apply)
2. Solve the various arithmetic operation in computer hardware (Apply)
3. Estimate the processor time and CPU usage. (Apply)
4. Design and analyze Memory devices (Analyze- for Assignment)

Reference Books:

1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill, 2002.
2. William Stallings: Computer Organization & Architecture, 9th Edition, Pearson, 2015.
3. M.Morris Mano, “Computer system Architecture”, 3rd Edition, Prentice-Hall Publishers, 2007.

ASSESSMENT STRUCTURE:

<table>
<tr><th rowspan="2">PCC</th><th rowspan="2">CIA</th><th rowspan="2">SEA</th><th colspan="3">CIA (50)</th><th rowspan="2">SEA
Conduction: 100 M
Reduced to: 50 M</th></tr>
<tr><td></td><td>I</td><td>II</td></tr>
<tr><td rowspan="5">Conduction</td><td rowspan="5">50</td><td rowspan="5">50</td><td rowspan="2">Written Test</td><td>50</td><td>50</td><td rowspan="4">Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module</td></tr>
<tr><td colspan="2">Average of two tests – 25 Marks</td></tr>
<tr><td>Assignment</td><td colspan="2">15</td></tr>
<tr><td>AAT</td><td colspan="2">10</td></tr>
<tr><td colspan="3"><b>Total – 50 marks</b></td><td><b>Total – 50 marks</b></td></tr>
</table>

i) CIA: 50%

IA Test: 2 IA tests - Each of 50 Marks	Average of 2 tests – scaled down to 25 M
Assignment – Two assignments – one for 10 marks and another for 5 marks	15 Marks
Additional Assessment Tools (AAT) – Oral /Online Quizzes, Presentations, Group discussions, Case studies, Term Paper, Open ended experiments, Mini industrial/social/rural Projects, Two-minute video on latest topic, Short MOOC courses, Practical Orientation on Design thinking, creativity & Innovation, Participatory & Industry integrated learning,	10 Marks

Practical activities, Problem solving exercises, Participation in seminars/academic events/symposia and any other activity	
Total	**50 Marks**

ii) SEA : 50%

Theory Exam	5 questions to answer each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – III

ARTIFICIAL INTELLIGENCE (PCI)

Credit : 3

Course Code	22AML133	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	1:2:2:0	SEA Marks	50
Total Number of Lecture Hours	40	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Learn the methods of solving problems using Artificial Intelligence.
- Learn the knowledge representation techniques, reasoning techniques and planning

	Number of Hours	**Bloom's Level**
Module-1 : Introduction		
Introduction to AI: history, Intelligent systems, foundation and sub area of AI, applications, current trend and development of AI. **Problem solving:** Production System, water jug problem, Missionaries and Cannibals Problem, 8-Puzzle problem, State space search, Control Strategies: Characteristics of Problem. **Practical:** 1. Write a Program to Implement Tic-Tac-Toe game using Python. 2. Write a Program to implement 8-Puzzle problem using Python. 3. Write a Program to Implement Water Jug using Python.	**6+4**	**(CO1) Apply**
Module-2 : Problem solving-1		
Uninformed Search Strategies: Breadth-First search, Uniform- Cost Search, Depth-first search, Depth-limited search, Iterative deepening depth-first search, Bidirectional search, comparing uninformed search strategies. **Informed (Heuristic) Search strategies:** Best-first search, A* algorithm, Memory-bounded Heuristic search-RBFS algorithm and SMA* algorithm, AO* algorithm **Practical:** 1. Implement AO* Search algorithm. 2. Write a Program to Implement Breadth First Search using Python. 3. Implement N-Queens problem	**6+4**	**(CO2) Apply**
Module-3 : Game Playing		

Adversarial Search: Nim Game problem, minimax procedure, alpha-beta pruning. **Constraint Satisfaction Problems:** Crypt-arithmetic problem **Advanced problem solving paradigm:** Planning: types of planning system, block world problem, logic based planning, Linear planning using a goal stack, sussman anomaly problem in goal stack, Means-ends analysis, Nonlinear planning strategies, learning plans. **Practical:** 1. Write a program to implement Missionaries and Cannibals. 2. Write a program to implement Monkey and Bananas Problem 3. Write a Program to Implement Tower of Hanoi	**6+4**	**(CO2) Apply**
Module-4 Logical Reasoning and planning		
Logical reasoning: propositional calculus, propositional logic, Natural Deduction system, Axiomatic system, Semantic Tableau system in propositional logic, resolution refutation in propositional logic, predicate logic, logic programming, Unification algorithm, forward and backward chaining, conflict resolution. **Practical:** 1. Implementation of the problem solving strategies: either using Forward Chaining or Backward Chaining. 2. Write predicates to convert centigrade temperatures to Fahrenheit and check if temperature is below freezing.	**6+4**	**(CO3) Apply**
Module-5: Knowledge Representation & Expert Systems		
Knowledge Representation: Approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, Knowledge representation using Frames. **Expert Systems:** Architecture of expert systems, Roles of expert systems - Knowledge Acquisition –Meta Knowledge. Typical expert systems - MYCIN, DART, XOON. **Practical:** 1. Implement MYCIN expert system	**6+4**	**(CO4) Apply**
Course outcomes: The students will able to 1. Understand the concepts of AI, characteristics of problems and apply various techniques for problem solving. 2. Apply appropriate search techniques to solve AI problems. 3. Develop knowledge base sentences using propositional logic and first order logic for logical reasoning. 4. Apply AI techniques for knowledge representation using semantic networks and implement various expert systems.		
Reference Books: 1. Staurt Russel, Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 3rd Edition, 2009 2. Elaine Rich, Kevin Knight, Artificial Intelligence, Tata McGraw Hill 3. George F Lugar, Artificial Intelligence Structure and strategies for complex, Pearson Education, 5th Edition, 2011 4. Saroj Kaushik, Artificial Intelligence, Cengage learning, 2014 5. Nils J. Nilsson, Principles of Artificial Intelligence, Elsevier, 1980		

ASSESSMENT STRUCTURE:

PCI	**CIA**	**SEA**	**CIA (50)**			**SEA** **Conduction: 100 M** **Reduced to: 50 M**
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 50 marks scaled down to 15 marks		
			Assignment	Average of 2 Assignments – 10M		
			Practical	Weekly Assessment – 10 Marks IA test – 15 Marks (IA test to be conducted for 50 M and scaled down to 15M)		
					Total – 50 Marks	**Total – 50 Marks**

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
	Total	**50 Marks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **Reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – III

Data Structures & Applications (PCI)
Credit: 4

Course Code	22AML134	CIA Marks	50
Teaching Hours/Week(L:T: P: J)	3:0:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Use and implement data structures to obtain solutions.
- Illustrate and implement data structures such as stack, queue and linked list and apply them for the given problem.
- Distinguish the conceptual and applicative differences in trees, binary trees and binary search trees.
- Create and use appropriate data structures in C programs for solving real life problems.

	Number of Hours	**Bloom's Level**
Module 1: Stacks and Queues		
Concept Learning: Data Structures: Classification (Primitive & Non-primitive), Operations, Pattern Matching Algorithms (Brute force, KMP) Stacks: Definition, Operations, Implementation using arrays, Applications of Stacks – Infix to Postfix Conversion and Postfix Expression Evaluation. Queues: Definition, Operations, Implementation, Applications, Circular Queue (Message queue using Circular queue), Doubly Ended Queue, Priority Queue. **Practical:** 1. Design, develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alpha numeric operands. 2. Design, Develop and Implement a Program in C to Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ using Stack. 3. Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) • Insert an Element on to Circular QUEUE • Delete an Element from Circular QUEUE • Demonstrate Overflow and Underflow situations on Circular QUEUE • Display the status of Circular QUEUE Support the program with appropriate functions for each of the above operations.	**6+4**	**(CO1, CO2, CO3) Apply**
Module 2: Linked List		
Linked Lists: Definition, Create, Insert, Delete, Update, Traverse, and Position-	**6+4**	**(CO1,**

based Operations, Concatenate, Merge, and Reverse Lists, Doubly Linked List Implementation and Operations, Circular Linked List Implementation and Operations, Applications of Lists (Polynomial addition). Implementation of stacks and queues using Linked List **Practical:** 1. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo. • Create a SLL of N Students Data by using front insertion. • Display the status of SLL and count the number of nodes in it • Perform Insertion / Deletion at End of SLL • Perform Insertion / Deletion at Front of SLL 2. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo • Create a DLL of N Employees Data by using end insertion. • Display the status of DLL and count the number of nodes in it • Perform Insertion and Deletion at End of DLL • Perform Insertion and Deletion at Front of DLL		**CO2, CO3) Apply**
Module 3: Trees		
Concept Learning: Trees: General Tree Representation, Traversals, Applications. Binary Trees: Definition, Properties, Traversals, Applications. Binary Search Tree: Definition, Implementation, Search, Insert, Delete operations. Building and Evaluating Binary Expression Tree. **Practical:** 1. Write a C program to find maximum depth or height and level of a full binary tree and complete binary tree. 2. Write a C program to print all the path from root to left path for given binary tree. 3. Write a C program to insert a new node as a left child in a threaded binary tree. 4. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers. • Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 • Traverse the BST in Inorder, Preorder and Post Order • Search the BST for a given element (KEY) and report the appropriate message	**6+4**	**(CO1, CO2, CO3) Apply**
Module 4: Advanced Trees & Hashing		
Concept Learning: Heap Tree: Definition, Implementation, Insert, Delete, Peek operations. Hashing: Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining. **Practical:** 1. Write a C program to construct MAX-Heap. 2. Write a C program to insert node AVL tree.	**6+4**	**(CO2,CO3, CO4) Apply**
Module 5: Graphs		
Concept Learning: Graphs: Disjoint sets, Representation of Graphs - Adjacency/ Cost Matrix, Adjacency Lists, and Traversal of Graphs (BFS and DFS) **Practical:** 1. Design, Develop and Implement a Program in C for the following	**6+4**	**(CO2, CO3, CO5) Apply**

operations on Graph(G) of Cities 2. Create a Graph of N cities using Adjacency Matrix. 3. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS 4. Write a C Program to detect Cycle in a Directed Graph 5. Write a C Program to find if there is a path between two vertices in a directed graph.		

Course outcomes:

The students will able to

1. Apply fundamental data structures viz., Lists, Stacks, Queues, Linked Lists, Binary Trees from first principles
2. Identify the use of appropriate data structures for a given problem.
3. Design and implement solutions to basic practical problems using customized data structures.
4. Apply the advanced concepts like Heap & Hashing to solve problems.
5. Apply the concepts to solve graphical problems.

Reference Books:

1. “Data Structures and Program Design in C”, Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla, PHI, 2nd Edition, 2015.
2. Y. Langasm, M. J. Augenstein, A. M. Tenenbaum (2001) Data Structures Using C and C++, Prentice Hall India, New Delhi, India.
3. T. H. Cormen, C. E. Leiserson and R. L. Rivest (1990) Introduction to Algorithms, Third Edition, MIT Press, MA.
4. Fundamentals of Data Structures in C -- by Horowitz, Sahni and Anderson-Freed (Silicon Press 2007).
5. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, 4th Revised edition; 2013, Addison-Wesley, ISBN-13: 978-8131714744
6. Data Structures Using C, Reema Thareja, 1st Edition, 2011, Oxford Higher Education, ISBN-13: 978-0198099307

ASSESSMENT STRUCTURE:

PCI	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M	
				I	II	PART A	PART B
Conduction	50	50	IA Test	30	30	30 Marks	70 Marks
				Average of two tests – 30 M			
			Continuous Assessment	Weekly Assessment -20 marks			
					Total – 50 Marks		**Total – 50 Marks**

i) CIA: 50%

IA Test: 2 IA tests - each of 30 Marks	Average of 2 tests – 30 M
Practical Lab record – 10 Marks Performance – 05 Marks Viva – 05 Marks	20 Marks
	Total 50 Marks

ii) SEA : 50%

Question Paper:

Theory part	5 questions to answer each of 6 Marks 2 questions from each module with internal choice Student should answer one full question from each module	6 M x 5 = 30 Marks
Execution part	Write up - 20 Marks Conduction - 40 Marks Viva-Voce - 10 Marks	70 Marks
	Total	**100 Marks** **Reduced to 50 M**

Note:

- No Assignment and AAT

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – III

MICROCONTROLLER AND EMBEDDED SYSTEMS (PCI)

Credit : 4

Course Code	22AML135	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	3:0:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Illustrate the logic design concepts and combinational logic circuits
- Provide the student with the basic understanding of microcontroller and embedded systems design.
- Learn the addressing modes, instructions, and assembler directives and develop the ALP to solve problems.
- Develop embedded C programs for microcontrollers and run on the simulator, target board and various interfaced hardware devices.
- Integrate Hardware and Software to Implement the required embedded smart systems

	Number of Hours	Bloom's Level
Module-1 Logic Design and Combinational Logic Circuits		
Logic Design & Applications: Basic Logic gates. **Combinational Logic Circuits:** Sum-of-Products Methods, Karnaugh Map simplifications, Don't – care Conditions, Product-of-Sums Simplifications, Simplification by Quine-McClusky Method, Map Entered Variable Method. **Practical:** Realize the following digital circuits using Digital Trainer kit. 1. Realization of Boolean expression: $Y= \bar{A}\,\bar{B}\,\bar{C}\,D+\bar{A}\,B\bar{C}\,D+AB\bar{C}\,D+\bar{A}\,B\,\bar{C}\,D+\bar{A}\,\bar{B}\,\bar{C}\,D+\bar{A}\,\bar{B}\,C\,\bar{D}+A\,B\,CD$ 2. Realize Half Adder, Full Adder, Half Subtractor and Full Subtractor using Logic Gates. 3. Realize Binary to Gray & Gray to Binary Code Converters using Logic Gates	**6+4**	**(CO1) Apply**
Module-2: ARM-32 bit Microcontroller		
ARM-32bit Microcontroller: Thumb-2 technology and applications of ARM, Architecture of ARM Cortex M3, Various Units in the architecture, debugging support, General Purpose Registers, Special Registers, exceptions, interrupts, stack operation, reset sequence. **Practical:** Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn ALP and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler. 1. Write a program to multiply two 16 bit binary numbers. 2. Write a program to find the sum of first 10 integer numbers. 3. Write a program to find factorial of a number. 4. Write a program to add an array of 16 bit numbers and store the 32 bit result in memory location.	**6+4**	**(CO2) Apply**
Module-3: ARM Cortex M3 Instruction Sets and Programming		

ARM Cortex M3 Instruction Sets and Programming: Assembly basics, Instruction list and description, Special instructions, Useful instructions, Assembly and C language Programming **Practical:** Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn ALP and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler. 1. Write a program to find the square of a number (1 to 10) using look-up table. 2. Write a program to find the largest/smallest number in an array of 32 numbers. 3. Write a program to arrange a series of 32-bit numbers in ascending/descending order. 4. Write a program to count the number of ones and zeros in two consecutive memory locations.	**6+4**	**(CO3) Apply**
Module-4: Embedded System Components		
Embedded System Components: Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems Core of an Embedded System, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Communication Interfaces (12C, SPI, IrDA, Bluetooth, Wi-Fi, Zigbee only). **Practical:** Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn ALP and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler. 1. Display “Hello World” message using Internal UART. 2. Interface and Control the speed of a DC Motor. 3. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction. 4. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.	**6+4**	**(CO4) Apply**
Module-5: Programming with Arduino		
Programming with Arduino: Understanding the ecosystem of Arduino, Pinout configuration, Digital input and output, Analog input and output, working with sensors and actuators, Arduino serial communication, Communication interfaces (SPI and I2C) communication. **Practical:** Conduct the following experiments by writing program using Arduino Uno board and the required software tool. 1. Interface a DHT11 sensor with Arduino Uno. 2. Interface GPS module with Arduino Uno. 3. Interface GSM module with Arduino Uno. 4. Interface LCD module with Arduino Uno.	**6+4**	**(CO5) Apply**
Course outcomes: The students will be able to 1. Apply Karnaugh Map, Quine-McClusky Method and Map Entered Variable Method to simplify digital circuits. (Apply) 2. Implement the architectural features and instructions of 32-bit microcontroller ARM CortexM3. (Apply) 3. Apply the knowledge gained for Programming ARM Cortex M3 for different applications. (Apply) 4. Apply the knowledge of basic hardware components and their selection method based on the characteristics and attributes of an embedded system. (Apply) 5. Interfacing various sensor modules with Arduino UNO using Arduino sketch to program the devices. (Apply)		
Reference Books: 1. M. Morris Mano, Digital Design, 4th Edition, Pearson Prentice Hall 2008 2. Joseph Yiu, "The Definitive Guide to the ARM Cortex-M3", 211d Edition, Newnes, (Elsevier), 2010.		

3. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education Private Limited, 2nd Edition.
4. Exploring Arduino: Tools and Techniques for Engineering, Wizardry 1st Edition WILEY, ISBN-10: 1118549368, ISBN-13: 978-1118549360.

ASSESSMENT STRUCTURE:

PCI	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 50 marks scaled down to 15 marks		
			Assignment	Average of 2 Assignments – 10M		
			Practical	Weekly Assessment – 10 Marks IA test – 15 Marks (IA test to be conducted for 50 M and scaled down to 15M)		
					Total – 50 Marks	**Total – 50 Marks**

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
	Total	**50 Marks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **Reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – III

Object Oriented Programming Using JAVA
Credit: 2

Course Code	22AML136	CIA Marks	50
Teaching Hours/Week(L:T: P: J)	0:0:2:2	SEA Marks	50
Total Number of Lecture Hours	30	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Understand and apply the basic concepts of object-oriented programming.
- Implement java programs for establish interfaces and to develop reusable software components.
- Build software development skills using java programming for real-world applications.

	Number of Hours	**Bloom’s Level**
Task 1		
An Overview of Java, Data Types, Variables, and Arrays, Operators, Control Statements, Classes and Methods 1. Write a JAVA program to display message “Welcome to BNMIT”and “I am first batch of Autonomous” in two different lines. 2. Write a JAVA program to display at-least five student informationby considering student USN, name, branch and semester. 3. Write a java program that prints all real solutions to the quadratic equation $ax^2 +bx+c=0$. Read in a, b, c and use the quadratic formula. 4. Write a java program to create objects of class Students with studentUSN, name, branch and semester and display information. 5. Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area().	**2**	**(CO1)** **Apply**
Task 2		
Method overloading, Inheritance, polymorphism, encapsulation 1. Calculate area of Rectangle, Triangle and Circle by using method overloading 2. Write a java program to create a class named Employee. Extend Faculty class from Employee class. Extend Professor Class from Faculty class. Access members of super class using super keyword. Create an instance to sub class called Professor and access members of both Faculty and Professor using instance. 3. Write a java program to develop a suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc. Design a simple test application to demonstrate dynamic polymorphism. 4. Write a java program to create a class Car that inherits from base class Vehicle using private strings and getter/setter methods to achieve encapsulation.	**2**	**(CO1)** **Apply**
Task 3		

Multithreaded Programming 1. Write a java program for multithread in which user thread and thread started from main method invoked at a time each thread sleep for 1 sec. 2. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number. 3. Write an application that executes two threads. One thread displays “An” every 1000 milliseconds and other displays “B” every 3000 milliseconds. Create the threads by extending the Thread class.	**2**	**(CO2) Apply**
Task 4		
Enumerations, Strings 1. Write a java program to create an enum of restaurants that can be used to pick user choice restaurant. 2. Given an input string, you are expected to extract either all vowels, or all non-vowels from the string and return the result as all lowercase or uppercase, based on the options specified. • input1 represents the input string. • input2 represents the extraction option. 0 for extraction of all non-vowels. 1 for extraction of all vowels. • input3 represents the output case option. 0 for all lowercase letters. 1 for all UPPERCASE letters. 3. Write a java program to find the duplicate words and their number of occurrences in a string. 4. Write a Java program to replace each substring of a given string that matches the given regular expression with the given replacement.	**2**	**(CO1) Apply**
Task 5		
Collections 1. Write a Java program to create a new array list, add some colors (string) and print out the collection. 2. Write a Java program to iterate through all elements in a linked list starting at the specified position. 3. Write a Java program to append the specified element to the end ofa hash set. 4. Write a Java program to create a new tree set, add some colors (string) and print out the tree set. 5. Write a Java program to create a new priority queue, add some colors (string) and print out the elements of the priority queue.	**2**	**(CO1) Apply**
Task 6		
Collections 1. Write a Java program to associate the specified value with the specified key in a Tree Map. 2. Write a Java program for the following: i) Create a doubly linked list of elements. ii) Delete a given element from the above list. iii) Display the contents of the list after deletion. 3. Write a Java program to store content in Hash table and use enumeration to display contents of Hash Table. 4. Write a Java program to create a vector of n elements and perform the following operations: Adding elements, Removing elementsand Display elements. Write a program to add elements to the HashMap given the key andvalue data type is string, get size of HashMap, and check if HashMap is empty.	**2**	**(CO1) Apply**

Task 7		
Event Handling 1. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.	**2**	**(CO2) Apply**
Task 8		
Event Handling 1. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event isfired.	**2**	**(CO2) Apply**
Task 9		
Event Handling 1. Write a java program that simulates a traffic light. The program lets the user select one of three lights: Red, Yellow or Green with radio buttons. On selecting a button an appropriate message with “STOP” or “READY” or” GO” should appear above the buttons in selected color. Initially, there is no message shown. 2. Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,*, % operations. Add a text field to display the result. Handle any possible exception like divided by zero.	**2**	**(CO2) Apply**
Task 10		
Event Handling 1. Write a java program to create a frame that contains two buttonsand one text field. 2. Write a java program to handle the button click events by implementing ActionListener Interface. 3. Write a java program to create two textfields to display single line text string and one TextArea that is used to display multiple-line text string. Both should be editable in nature. 4. Write a java program to create a drop-down menu of choices. Whena user selects a particular item from the drop-down then it is shown on the top of the menu.	**2**	**(CO2) Apply**
Task 11		
Exception Handling 1. Write a java program to represent a list of items together and popup menu to display some message. One or more than one item can be selected from the list. 2. Write a program in java if number is less than 10 and greater than 50 it generates the exception out of range. Else it displays the square of number. 3. Write a program in java to enter the number through command line argument. If first and second number is not entered then it will generate the exception. Also divide the first number with second number and generate the arithmetic exception.	**2**	**(CO2) Apply**
Task 12		

Java Script 1. Write a program to Swap Two Variables. 2. Write a program to Generate a Random Number. 3. Write a program to Check the Number of Occurrences of aCharacter in the String. 4. Write a program to Count the Number of Vowels in a String. 5. Write a java script program to pass a 'javascript function' asparameter.	**2**	**(CO3) Apply**
Task 13		
File handling 1. Write a java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes. 2. Write a java program that displays the number of characters, lines and words in a text file.	**2**	**(CO2) Apply**
Task 14		
File handling 1. Write a java program that reads a file and displays the file on the screen with line number before each line. 2. Write a java program in which data is read from one file and should be written in another file. Name of both file is given through command line arguments. 3. Write a java program in which data is read from one file and should be written in another file line by line.	**2**	**(CO2) Apply**
Task 15		
Mini-Project • Develop real world application using graphical user interface and object-orient concept for selected problem statement. • The problem statement can be selected from the following title but not limitedto the same. 1. Electricity bill generation 2. Currency converter / Distance converter / Time converter 3. Pay slip generation 4. Online book store 5. Airline reservation system 6. Designing of simple calculator		**(CO4) Create**

Course Outcomes:

The students will be able to

1. Develop Java application programs to implement basic Object Oriented concepts. (Apply)
2. Apply the concepts of Multithreading, Exception handling, event and file handling to develop efficient and error free codes. (Apply)
3. Design event driven GUI and web related applications which mimic the real word scenarios. (Apply)
4. Design, implement, test, and debug graphical user interfaces to solve real time applications.(Create-for Mini project).

Reference Books:

1. Herbert Schildt, “Java, The complete Reference”, Tata McGraw-Hill, 7^{th} Edition.
2. P. J. Deitel, H. M. Deitel, “Java for Programmers”, Pearson Education, PHI, 4th Edition,2007.
3. P. Radha Krishna, “Object Oriented Programming through Java”, Universities Press, 2ndEdition, 2007
4. Bruce Eckel, “Thinking in Java”, Pearson Education, 4th Edition, 2006.
5. Sachin Malhotra, Saurabh Chaudhary, “Programming in Java”, Oxford University Press, 5thEdition, 2010.

Assessment Structure:

PBL	**CIA**	**SEA**	**CIA (50)**			**SEA** **Conduction: 100 M** **Reduced to: 50 M**
Conduction	50	50	Theory	**I IA**	**II IA**	Project Assessed for 100 marks reduced to 50 Marks
				25	25	
				Average of 2 tests – 25 M		
			Practical	Weekly Assessment (Record/Project) – 10 Marks Lab IA test – 15 Marks		
					Total – 50 Marks	**Total – 50 Marks**

i) CIA: 50%

Theory - 2 IA tests - Each of 25 Marks	25 Marks
Practical **Weekly Assessment** - Lab record/Project – 10 Marks **Lab IA test –** 15 Marks	25 Marks
Total	**50 Marks**

ii) SEA : 50%

Project	Write up – 10 Marks Project report – 25 Marks Presentation & Demonstration - 50 Marks Viva-Voce – 15 Marks	100 Marks Reduced to 50 Marks
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Mathematics

Syllabus

Semester: IV		
Course: Statistics, Probability and Graph theory **Course Code: 22MAI141 (Common to CSE, ISE, AIML)**		
L:T:P:J	**2:2:0:0**	**CIA: 50**
Credits:	**03**	**SEA: 50**
Hours:	**40**	**SEA Duration: 03 Hours**

Course Learning Objectives: The students will be able to	
1	Provide an insight into applications of Graph Theory, Curve fitting & Statistical methods.
2	Develop the knowledge of probability, joint probability distribution and Queuing theory occurring in digital signal processing, design engineering and micro wave engineering.

Module-1: Curve fitting & Statistical methods	**No. of hours**	**Blooms cognitive Levels**
Examples from Engineering that require curve fitting and statistical methods. **Curve Fitting:** Curve fitting by the method of least squares-fitting the curves of the form: $y = ax+b$, $y = ax^b$ *and* $y = ax^2 + bx + c$. **Statistical methods:** Introduction to Moments, Skewness, Kurtosis and problems. Karl Pearson's coefficient of correlation and lines of regression. *Experiential Learning component: Problems on curve fitting and statistical methods*	**L: 04** **T: 04**	**Apply**
Module-2: Probability distributions & Joint probability distribution		
Examples from Engineering that require Probability and Joint probability distribution **Probability distributions:** Review of basic probability theory. Discrete and continuous Random variables, probability mass/density functions (definitions only). Binomial, Poisson, exponential and normal distributions (without proof). **Joint probability distribution:** Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. *Experiential Learning component: Problems on Binomial, Poisson, Exponential and Normal distributions*	**L: 04** **T: 04**	**Apply**
Module-3:Markov chain & Sampling theory		
Examples from Engineering that require Markov Chain and Sampling Theory **Markov chain**: Introduction to Stochastic process, Probability vectors, Stochastic matrices, Regular stochastic matrices, Markov Chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states, Markovian processes. **Sampling theory**: Introduction to sampling theory, testing of hypothesis, level of significance, confidence limits, test of significance of mean and difference of means for large samples-z-test, test of significance of small Samples-Student's t- distribution, Goodness of fit-Chi-Square test. *Experiential Learning component: Problems on Markovian processes and, Sampling Theory*	**L: 04** **T: 04**	**Apply**
Module-4: Queuing theory		
Examples from Engineering that require queueing theory Introduction, birth and death process, Kendall's Notation, Symbolic representation of a queuing model, single server Poisson queuing model with infinite capacity (M/M/1: ∞/FCFS), when $\lambda_n = \lambda$ and $\mu_n = \mu(\lambda < \mu)$, Multiple server Poisson queuing model with infinite capacity (M/M/S: ∞/ FCFS), when $\lambda_n = \lambda$ for all $n, (\lambda > S\mu)$, *Experiential Learning component: Problems on* (M/M/1: ∞/FCFS) and (M/M/S: ∞/ FCFS) queuing models	**L: 04** **T: 04**	**Apply**
Module-5: Graph theory		
Examples from Engineering that require graph theory Basic concepts, types of graphs, order and size of a graph, in-degree and out-degree, bipartite-graphs, connected and disconnected graphs, Eulerian graph, Hamiltonian graphs, sub-graphs, isomorphic graphs. Matrix representation of graphs, adjacency matrix, incidence matrix. Planar graphs: definition, characterization of planar graphs, Kuratowski's theorem, Euler's formula and consequences. *Experiential Learning component: Problems on detection of planar and non-planar graphs*	**L: 04** **T: 04**	**Apply**

Course Outcomes: After completing the course, the students will be able to

CO 1: Make use of correlation and regression analysis to fit a suitable mathematical model for the statistical data.

CO 2: Apply discrete and continuous probability and joint probability distributions in analyzing the probability models arising in engineering field.

CO 3: Use Markov chain in prediction of future events and demonstrate the validity of testing the hypothesis.

CO 4: Acquire skills in analyzing queuing models.

CO 5: Apply the knowledge of Graph Theory in Network modeling, electrical network and computational algorithms.

Reference Books:

1. E. Kreyszig: “Advanced Engineering Mathematics”, John Wiley & Sons, 10th Edition(Reprint), 2016.
2. B. S. Grewal: “Higher Engineering Mathematics”, Khanna Publishers, 44th Edition, 2017.
3. S. D. Sharma : “Operations Research", Kedar Nath Ram Nath & Co. Meerut, 2014.
4. T. Veerarajan : Probability, Statistics and Random processes, McGraw Hill Education(India) Private Limited, Third edition, Nineteenth reprint 2017.
5. C. Ray Wylie, Louis C. Barrett : “Advanced Engineering Mathematics", 6th Edition, McGraw-Hill Book Co., New York, 1995.
6. James Stewart : Calculus-Early Transcendental, Cengage Learning India Private Ltd., 2017.
7. B. V. Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
8. Srimanta Pal & Subodh C. Bhunia: “Engineering Mathematics”, Oxford University Press, 3rd Reprint, 2016.
9. Narsingh Deo, “Graph Theory with Applications to Engineering and Computer Science”, Prentice Hall of India, 2000.

Web links and Video Lectures:

1. https://nptel.ac.in/courses/111104098
2. https://www.youtube.com/watch?v=1YkfeR05YXY
3. https://archive.nptel.ac.in/courses/111/104/111104079/
4. https://www.youtube.com/watch?v=xGkpXk-AnWU
5. https://archive.nptel.ac.in/courses/106/104/106104170/

ASSESSMENT STRUCTURE:

PCC	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 25 Marks		
			Assignment	15		
			AAT	10		
			Total – 50 marks			Total – 50 marks

i) CIA: 50%

IA Test: 2 IA tests - Each of 50 Marks	Average of 2 tests – scaled down to 25 M
Assignment – Two assignments – one for 10 marks and another for 5 marks	15 Marks
Additional Assessment Tools (AAT) – Oral /Online Quizzes, Presentations, Group discussions, Case studies, Term Paper, Open ended experiments, Mini industrial/social/rural Projects, Two-minute video on latest topic, Short MOOC courses, Practical Orientation on Design thinking, creativity & Innovation, Participatory & Industry integrated learning, Practical activities, Problem solving exercises, Participation in seminars/academic events/symposia and any other activity	10 Marks
Total	**50 Marks**

ii) SEA : 50%

Theory Exam	5 questions to answer each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M reduced to 50 M**
Total		**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

Operating Systems (PCC)

SEMESTER–IV

Subject Code: **22AML142**	L:T:P:J: 2:1:1:0	CIA Marks:	50
Credits:	3	SEA Marks:	50
Total Number of Lecture Hours	40	SEA Duration:	3 Hours

Course Objectives:

- Introduce concepts and terminology used in OS
- Explain threading and multithreaded systems
- Illustrate process synchronization and concept of Deadlock
- Introduce to Unix File Systems

Module 1: Introduction to Operating System & Process Management	**Teaching Hours**	**Blooms cognitive Levels**
Fundamental Concepts of Operating System: Introduction to Operating systems, Operating system functions and services, historical evolution of operating systems, System boot. **Process Management:** Process abstraction, process address space, process management, system calls, threads. **CPU Scheduling:** Levels of scheduling, comparative study of scheduling algorithms, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling, Multi-processor scheduling.	**8**	**(CO1) Apply**
Module 2: Process Synchronization and Deadlocks		
Concurrent Processes: Critical section problem, semaphores, Classical problems of synchronization, monitors, inter-process communication, message passing mechanisms. **Deadlocks:** Characterization, prevention and avoidance, deadlock detection and recovery.	**8**	**(CO2) Apply**
Module 3: Memory Management		
Memory Management: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation, virtual memory concept, demand paging, page replacement algorithms, thrashing, Disk Scheduling. **Solid State Drives**-SSD Architecture, Flash Controller, Garbage Collection, Bad Block Management.	**8**	**(CO3) Apply**
Module 4: UNIX file system		
Unix files: UNIX Architecture, Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. **Practical component**: Execution of UNIX Shell Commands	**8**	**(CO4) Apply**
Module5: File attributes and Shell programming		
File attributes and permissions: The ls command with options. Changing file	8	**(CO5) Apply**

permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. The shells interpretive cycle: Wild cards. Removing the special meanings of wild cards. Three standard files and redirection. **Connecting commands**: Pipe, grep, egrep. **Shell programming**: Ordinary and environment variables. Read and read-only commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document. Simple shell program examples. **Practical component:** Execution of Wildcards & UNIX Shell Programs		

Course Outcomes: After completing the course, the students will be able to

1. Apply the concepts of process scheduling to improve CPU utilization and identify various multi-threading models
2. Identify the need of policies, protection required in managing deadlock, main and virtual memory & various techniques in managing concurrent processes.
3. Apply the concept of paging & segmentation for effective memory management.
4. Apply the concepts of Unix system and file commands to perform various tasks in files and system.
5. Analyze the concepts of Wildcards and Shell Programming to write basic shell scripts and formulating regular expressions for Pattern matching

Reference Books:

1. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill.
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006
3. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 9th Edition, 2018.
3. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005
4. Unix System Programming Using C++ - Terrence Chan, PHI, 1999.

Web links and Video Lectures:

1. https://academicearth.org/
2. https://archive.nptel.ac.in/courses/106/105/106105214/

ASSESSMENT STRUCTURE:

PCC	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 25 Marks		
			Assignment	15		
			AAT	10		
				Total – 50 marks		**Total – 50 marks**

i) CIA: 50%

IA Test: 2 IA tests - Each of 50 Marks	Average of 2 tests – scaled down to 25 M
Assignment – Two assignments – one for 10 marks and another for 5 marks	15 Marks
Additional Assessment Tools (AAT) – Oral /Online Quizzes, Presentations, Group discussions, Case studies, Term Paper, Open	10 Marks

	ended experiments, Mini industrial/social/rural Projects, Two-minute video on latest topic, Short MOOC courses, Practical Orientation on Design thinking, creativity & Innovation, Participatory & Industry integrated learning, Practical activities, Problem solving exercises, Participation in seminars/academic events/symposia and any other activity	
	Total	**50 Marks**

ii) SEA : 50%

Theory Exam	5 questions to answer each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

Semester: IV

Database Management System (PCI)
Credit: 3

Course Code:	22AML143	**CIA Marks:** 50
Credits:	**3:0:1:1**	**SEA Marks:** 50
Total Number of Lecture Hours	**50**	**SEA Duration:** 03

Course Learning Objectives: The students will be able to

Understand fundamental concepts, terminology and application of databases, SQL and NoSQL

Design concepts and creation of relational databases using relation algebra.

Practice SQL programming through a variety of database problems.

Demonstrate the use of Normalization, concurrency and transactions in database.

Module-1: Database System Concepts, Data Modeling	**No. of Hours**	**Blooms cognitive Levels**
Databases and Databases Users: Characteristics of database Approach, Advantages of using the DBMS Approach. Database System Concepts and Architecture: Data Models-Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces. Data Modeling Using the Entity-Relationship (ER) Model: Entity Types-Entity sets- Attributes and Keys, Relationship types – Relationship Sets – Roles and structural Constraints, Weak Entity Types. **Practical component:** Draw ER Diagram for the following Databases using GitMind software. Order Database Library Database Bank Database	10	**Understand** **CO1**
Module-2: Relational Data Model and Relational Algebra		
Concept of relations, schema-instance distinction, keys, referential integrity and foreign keys, relational algebra operators: selection, projection, cross product, various types of joins, division, example queries, tuple relation calculus, domain relational calculus, converting the database specification in E/R notation to the relational schema **Practical component:** Create Schema, insert at least 5 records in each table and add appropriate constraints for the following Library Database using ORACLE or MySQL DBMS under LINUX/Windows environment BOOK (Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS (Book_id, Author_Name)	**10**	**Apply** **CO2**

PUBLISHER (Name, Address, Phone) BOOK_COPIES (Book_id, Branch_id, No-of_Copies) BOOK_LENDING (Book_id, Br_id, Card_No, Date_Out, Due_Date) LIBRARY_BRANCH (Branch_id, Branch_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher,authors, number of copies in each branch, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books,but from Jan 2020 to Jun 2022. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.		
Module-3: SQL		
Basic SQL: SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT – DELETE and UPDATE Statements in SQL, Additional features in SQL **More SQL: Complex Queries, Triggers, Views and Schema Modification:** Complex SQL Retrieval Queries, Specifying Constraints as Assertions and actions as Triggers, Views (Virtual Tables) in SQL. **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Order Database using ORACLE or MySQL DBMS under LINUX/Windows environment. SALESMAN (Salesman_id, Name, City, Commission) CUSTOMER (C_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, C_id, S_id) Write SQL queries to 1. Count the customers with grades above Bangalore’s average. 2. Find the name and numbers of all salesman who had more than onecustomer. 3. List all the salesman and indicate those who have and don’t have customers in their cities (Use UNION operation.) Create a view that finds the salesman who has the customer with the highest order.	**10**	**Apply CO3**
Module-4: Functional Dependencies and Normalization		
Basics of Functional Dependencies and Normalization for Relational Database: Functional Dependencies, Armstrong’s axioms for FD's, Equivalent Decompositions, closure of a set of FDs, minimal covers, Normal forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce- Codd Normal Forms [BCNF] **Practical component:** Create Schema, insert at least 5 records for each table and add appropriate constraints for the following Company Database using ORACLE or MySQL DBMS under LINUX/Windows environment. EMPLOYEE (SSN, Name, Address, Sex, Salary, Super SSN, D No) DEPARTMENT (D No, D Name, Mgr. SSN, Mgr. Start Date) DLOCATION(D No,D Loc)	**10**	**Analyze CO4**

PROJECT (P No, P Name, P Location, D No) WORKS_ON(SSN, P No, Hours) Write SQL queries to 1. 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. 2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary and the average salary in this department.		
Module-5: Transaction Processing, Concurrency Control, NoSQL		
Introduction to Transaction Processing –Introduction to Transaction Processing, Desirable Properties on Transactions (ACID) **Concurrency Control Techniques:** Transactions and Schedules, Serializability, Precedence Graphs, Concurrency, Lock Based Protocols: 2PL, Strict 2PL Protocols, Deadlocks - Detection and Prevention **NoSQL:** SQL v/s NoSQL, The Emergence of NoSQL, BASE Properties, Data Models: Relationships, Graph Database, Schema less Database.	**10**	**Analyze CO5**

Course Outcomes: After completing the course, the students will be able to

1. Understand the Database System Concepts along with Data Modeling Using the Entity-Relationship (ER) Model
2. Apply the concepts of relations on RDBMS, constraints, joints using relational algebra operators.
3. Apply Structured Query Language for database manipulation.
4. Analyze functional dependencies to normalize relations of relational database.
5. Analyze transactions processing, schedules protocols, serializability issues, deadlocks in DBMS and concepts of NoSQL with its advantages

References:

1. Ramez Elmasari, Shamkant B Navathe “Fundamentals of Database Systems”, Pearson, Seventh Edition 2017.
2. “Database System Concepts”, Silberschatz, H Korth, S Sudarshan, 6th Edition, McGraw-Hill, 2010
3. Pramod J Sadalage, Martin Fowler, “NOSQL Distilled”, Pearson, 2013

ASSESSMENT STRUCTURE:

PCI	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 50 marks scaled down to 15 marks		
			Assignment	Average of 2 Assignments – 10M		
			Practical	Weekly Assessment – 10 Marks IA test – 15 Marks (IA test to be conducted for 50 M and scaled down to 15M)		
					Total – 50 Marks	**Total – 50 Marks**

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
	Total	**Marks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **Reduced to 50 M**
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – IV

DESIGN AND ANALYSIS OF ALGORITHMS
Credit : 4

Course Code	21AML144	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	3:0:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:
This course will enable students to

- Explain various computational problem solving techniques.
- Apply appropriate method to solve a given problem.
- Describe various methods of algorithm analysis.

	Number of Hours	**Bloom's Level**
Module-1: Introduction		
Introduction: Notion of algorithm, Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithmic Efficiency: Analysis frame work, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms. **Practical:** 1. Implement Coin Changing problem method and find the time required. 2. Write a program to find maximum profit using Knapsack technique. 3. Implement Job Sequence problem using Greedy method.Apply brute force/divide and conquer technique to recursively implement the following concepts: a. Linear Search or Binary Search. b. To find the maximum and minimum from a given list of n elements using Brute Force Method. 4. There are 5 books in the shelf, find the number of ways to select 3 books from 5 books on the shelf using the NCR with recursion. 5. Find the next three terms of the sequence 15, 23, 38, 61, … Fibonacci series of the given number using recursion. 6. Demonstrate through a program how a sequence of characters is taken and checked for the possibility of the presence of the required string. If the possibility is found then, character matching is performed else no using Rabin Karp method.	**6+4**	**(CO1) Apply**
Module-2: Brute Force		
Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute		

Force String Matching Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the Maximum and Minimum, Merge sort, Quick sort, Strassen’s matrix multiplication. Decrease and Conquer Approach: Topological Sort. **Practical:** 1. Implement the Selection sort algorithm. 2. Implement Bubble sort algorithm. 3. Implement the Sequential Search algorithm. 4. Implement the String Matching algorithm. 5. Write a program to search a key in a given set of elements using Binary search method and find the time required to find the key. 6. Write a program to find Maximum and Minimum using divide and conquer technique and find the time required to find the elements. 7. Sort a given set of elements using Merge Sort method and determine the time required sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm. 8. Sort a given set of elements using Quick Sort method and determine the time required sort the elements. Plot a graph of number of elements versus time taken. Specify the time efficiency class of this algorithm. 9. Implement Topological sort using source removal method find the time required to sort the elements.	**6+4**	**(CO1) Apply**
Module-3: Greedy Method		
Greedy Method: General method, Coin Change Problem, 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. **Practical:** 1. Implement Coin Changing problem method and find the time required. 2. Write a program to find maximum profit using Knapsack technique. 3. Implement Job Sequence problem using Greedy method. 4. Implement Prim’s algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 5. Implement Kruskal’s algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 6. Implement Dijkstra’s algorithm find shortest paths to other vertices from a given vertex in a weighted connected graph.	**6+4**	**(CO2) Apply**

Module-4: Dynamic Programming		
Dynamic Programming: General method with Examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem, Bellman-Ford Algorithm, Travelling Sales Person problem, Reliability design. **Practical:** 1. Implement all-pairs shortest paths problem using Floyd's algorithm. 2. Implement all-pairs shortest paths problem using Warshal's algorithm. 3. Implement 0/1 Knapsack using Dynamic Programming. 4. Implementation of Bellman Ford Algorithm using a directed graph. 5. Implement Travelling Sales man problem using Dynamic Programming.	**6+4**	**(CO2) Apply**
Module-5: Backtracking		
Backtracking: General method (T2:7.1), N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. Programme and Bound: Assignment Problem, Travelling Sales Person problem, 0/1 Knapsack problem: LC Programme and Bound solution, FIFO Programme and Bound solution. NP- Complete and NP-Hard problems: Basic concepts, nondeterministic algorithms, P, NP, NP-Complete and NP-Hard classes **Practical:** 1. Implementation of N Queen Problem using Backtracking technique. 2. Implementation of SUM-SUBSET Problem. 3. Design and implement to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using the backtracking principle. 4. Implementation Assignment Problem using Backtracking technique. 5. Implementation of Travelling Sales Man Problem using Branch and Bound method. 6. Implementation of Knapsack problem using Branch and Bound method.	**6+4**	**(CO2) Apply**
Course outcomes: The students will able to 1. Understand the fundamentals of algorithms and develop an algorithm using appropriate design strategies for problem solving (Apply) 2. Use various design techniques such as dynamic programming, greedy algorithm and back tracking to design algorithms for more complex problems and analyze their performance. (Apply) 3. Estimate the computational complexity of different algorithms. (Apply) 4. Analyze computational complexity of an algorithm to increase efficiency. (Analyze)		
Reference Books: 1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson, 2nd Edition, 2009]		

2. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, Introduction to Algorithms, PHI, 3rd Edition
3. Ellis Horowitz, Satraj Sahni and Rajasekaran, Computer Algorithms/C++, Universities Press, 2nd Edition, 2014.

ASSESSMENT STRUCTURE:

PCI	**CIA**	**SEA**	**CIA (50)**			**SEA** **Conduction: 100 M** **Reduced to: 50 M**	
				I	II	**PART A**	**PART B**
Conduction	50	50	IA Test	30	30	30 Marks	70 Marks
				Average of two tests – 30 M			
			Continuous Assessment	Weekly Assessment -20 marks			
			Total – 50 Marks			**Total – 50 Marks**	

i) CIA: 50%

Test: 2 IA tests - each of 30 Marks	Average of 2 tests – 30 M
Practical Lab record – 10 Marks Performance – 05 Marks Viva – 05 Marks	20 Marks
	Total 50 Marks

ii) SEA : 50%

Question Paper:

Theory part	5 questions to answer each of 6 Marks 2 questions from each module with internal choice Student should answer one full question from each module	6 M x 5 = 30 Marks
Execution part	Write up - 20 Marks Conduction - 40 Marks Viva-Voce - 10 Marks	70 Marks
	Total	**100 Marks** **Reduced to 50 M**

Note:

- No Assignment and AAT

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – IV

MACHINE LEARNING (PCI)
Credit: 3

Course Code	**21AML145**	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	1:2:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Define Machine learning and understand the basic theory underlying machine learning.
- Differentiate supervised and unsupervised learning
- Understand the Linear Models for Regression and Classification
- Understand the basic concepts of learning and decision trees.
- Understand Bayesian techniques for problems appear in machine learning.

	Numbe rof Hours	**Bloom's Level**
Module-1		
Machine learning Landscape: ML Concepts, Uses of ML, Types of ML, Main challenges of ML **End to end Machine learning Project:** Working with real data, Look at the Big picture, Get the data, Discover and visualize the data, Prepare the data for ML Algorithm, Select and train the model, Fine tune your model. **Practical:** 1. Demonstrate Preprocessing (Data Cleaning, Integration and Transformation) activity on suitable data: For example: Identify and Delete Rows that Contain Duplicate Data by considering an appropriate dataset. Identify and Delete Columns That Contain a Single Value by considering an appropriate dataset.	**10**	**(CO1)** **Apply**
Module-2		
Concept learning and Learning Problems: Concept Learning Task – Find S - Version Spaces and Candidate Elimination Algorithm. **Classification:** MNIST, training a Binary classifier, performance measure, multiclass classification, error analysis, multi label classification, multi output classification. **Practical:** 1. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm. Output a description of the set of all hypotheses consistent with the training examples. 2. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file and show the output for test cases. Develop an interactive program by comparing the result by implementing LIST THEN ELIMINATE algorithm.	**10**	**(CO2)** **Apply**

Module-3		
Linear Models for Regression: Linear Regression, Polynomial Regression, Logistic Regression **Linear Models for Classification:** Training a Binary Classifier, Performance Measures, Measuring Accuracy Using Cross-Validation, Confusion Matrix, Precision and Recall, Tradeoff. **Practical:** 1. Write a python program to predict home prices using Linear Regression. 2. Write a python program to predict the weather using parameters with Linear Regression	**10**	**(CO3)** **Apply**
Module-4		
Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate Problem For Decision Tree Learning, The basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Learning, Issues in Decision Tree Learning **Ensemble learning and Random Forest** Voting Classifiers, Bagging and Pasting, Random Patches, Random Forests, Boosting, Stacking. **Practical:** 1. Implement an automated customer information system to direct the customer to correct department based on preference using Decision Trees 2. Write a python program to decide whether the budget of a company is exceeding or not with decision trees, with a sample dataset	**10**	**(CO5)** **Apply**
Module-5		
Bayesian Learning: Introduction, Bayes theorem, Bayes theorem and Concept Learning, Naïve Bays Classifier, Bayesian Belief Network, EM Algorithm. **Practical:** 1. Implement Bayes' Theorem for the following problem statement: In XYZ University, admissions for the departments IT, CSE, AIML are 75%, 85%, and 90% respectively in the previous year. In the total of their output 5, 3, 2 percent are slow learners students. A student is taken at random from the department and is found to be slow learners. What are the probabilities that it was from the department IT, CSE, AIML?	**10**	**(CO5)** **Apply**
Course outcomes: The students will able to: 1. Apply the concepts of Machine Learning and theory underlying machine learning. (Apply) 2. Apply the data to understand the distribution of the data. (Apply) 3. Apply Linear Models for Regression and Classification to classify the data. (Apply) 4. Demonstrate the working of various algorithms with respect to training and test data sets. (Apply) 5. Analyze the problems on Decision tree, Bayesian and Instant learning techniques. (Apply)		
Reference Books: 1. Aurelien Geron, "Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow",O'Reilly 2019. 2. Tom M. Mitchell, "Machine Learning", McGraw Hill Education (India) Edition, 2013.		

ASSESSMENT STRUCTURE:

PCI	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
				I	II	
Conduction	50	50	Written Test	50	50	Five questions with each of 20 marks (with internal choice). Student should answer one full question from each module
				Average of two tests – 50 marks scaled down to 15 marks		
			Assignment	Average of 2 Assignments – 10M		
			Practical	Weekly Assessment – 10 Marks IA test – 15 Marks (IA test to be conducted for 50 M and scaled down to 15M)		
			Total – 50 Marks			**Total – 50 Marks**

i) CIA: 50%

Theory	**IA Test** (Theory): 2 IA tests - each of 50 Marks – Average of 2 tests scaled down to 15 Marks **Assignment** : 2 Assignments – each of 10 marks	25 Marks
Lab	**Weekly Assessment** – 10 Marks **Practical test (1)** - 15 marks	25 Marks
Total		**50 Marks**

ii) SEA : 50%

Question Paper:

Theory Exam	5 questions to answer, each of 20 Marks 2 questions from each module with internal choice Student should answer one full question from each module	20 M x 5 = **100 M** **Reduced to 50 M**
Total		**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – IV

Python Programming and Applications (PBL)

Credit: 2

Course Code	22AML146	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	0:0:2:2	SEA Marks	50
Total Number of Lecture Hours	30	Exam Hours	03

Course Learning Objectives:

This course will enable students to

1. Acquire knowledge and programming skills in python to solve problems in various domains using different tools.
2. Understand the representation and use of primitive data types, operators, control structure, and built-in data structures.
3. Develop the ability to write database applications and perform file handling, exception handling and using Python.
4. Develop Graphical user interfaces and develop application to read/write data from/to files in python.

	Number of Hours	**Bloom's Level**
Task – 1		
Python introduction, data types, operators, flow control and exception handling in Python	**2**	**(CO1) Apply**
• Write a program to demonstrate different number datatypes in python. • Write a program to perform different arithmetic operations on numbers in python. • Write a python program to find the square root. • Write a python program to calculate the area of a triangle. • Write a python program to generate a random number. • Write a python program to find largest of three numbers. • Write a python program to print a number is positive/negative using if-else.		
Task – 2		
Functions, passing parameters and return values	**2**	**(CO1) Apply**
• Write a python program to find factorial of a given number using functions • Write a program to double a given number and add two numbers using lambda() • Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program		

which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed. • Develop a python program to convert binary to decimal, octal to hexadecimal using functions.		
Task – 3		
String Related Operations	**2**	**(CO1)** **Apply**
• Write a python program to create, concatenate and print a string • Write a Python program to print substring from a given string. • Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters. • Write a Python program to find the string similarity between two given strings		
Task – 4		
Lists, List Processing, Tuples, and Dictionaries.	**4**	**(CO2)** **Apply**
a. Write a python program to print duplicates from a list of integers and remove them from the list. b. Write a python program to create a list and perform the following methods 1) insert() 2) remove() 3) append() 4) len() 5) pop() 6) clear(). c. Write a python program to create a tuple and perform the following methods 1) Add items 2) len() 3) check for item in tuple 4)Access iems d. Write a python program to create a dictionary and apply the following methods 1) Print the dictionary items 2) access items 3) use get() 4)change values 5) use len() e. Write a python program that takes two lists and returns True if they are equal otherwise false		
Task – 5		
Pattern Matching with Regular Expression	**2**	**(CO3)** **Apply**
• Write a function called isphonenumber () to recognize a pattern 415-555-4242 without using regular expression and also write the code to recognize the same pattern using regular expression. • Develop a python program that could search the text in a file for phone numbers (+919900889977) and email addresses (sample@gmail.com) • Write a python program to match parenthesis in the given equation. • Write a Python program to match string using regular expression.		
Task – 6		
File Handling	**4**	**(CO3)** **Apply**
• Write a python program to open and write “hello world” into a file? • Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file? • Write a python program to open a file and check what are the access permissions acquired by that file using os module? • Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first the second file.		

• Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order		
Task – 7		
Classes, methods, objects, inheritance, polymorphism, overriding	**4**	**(CO4) Apply**
• By using the concept of inheritance write a python program to find the area of triangle, circle and rectangle. • Write a python program by creating a class called Employee to store the details of Name, Employee_ID, Department and Salary, and implement a method to update salary of employees belonging to a given department. • Write a python program to find the whether the given input is palindrome or not (for both string and integer) using the concept of polymorphism and inheritance.		
Task – 8		
Working with excel spreadsheets and web scraping	**4**	**(CO5) Apply**
• Demonstrate python program to read the data from the spreadsheet and write the data in to the spreadsheet • Write a Program to append data into spreadsheet. • Write a python program to download the all XKCD comics		
Task – 9		
Working with PDF, word and JSON files, Sending Email and Text Messages	**4**	**(CO5) Apply**
• Write a python program to combine select pages from many PDFs • Write a python program to fetch current weather data from the JSON file. • Write a Python program to send e-mail, • Write a Python program to send message updates.		
Task – 10		
Image Processing	**2**	**(CO5) Apply**
• Write a python program to perform open(), show(), rotate() and convert to grayscale image by processing any image. • Write a python program to print thumbnails, resize() the image by processing any image. • Write a python program to convert an image to ASCII image in Python. • Write a python program to plot solar image and flipping any image. • Write a python program to load an image in grayscale mode. By grayscale mode, convert this image to a black & white image composing by shades of gray and count white dots on a black background.		
Mini Project		
• Develop real world application using Python for selected problem statement. • The problem statement can be selected from the following title but not limited to the same. Temperature converter tool		

Stock investment tracker Election simulator High-scores tracker Dice Rolling Simulator	

Course outcomes:

The students will be able to

1. Demonstrate proficiency in handling of loops and creation of functions.
2. Identify the methods to create and manipulate lists, tuples and dictionaries.
3. Discover the commonly used operations involving regular expressions and file system.
4. Interpret the concepts of Object-Oriented Programming as used in Python.
5. Determine the need for scraping websites and working with PDF, JSON and other file formats.

Reference Books:

1. Al Sweigart, “Automate the Boring Stuff with Python”,1stEdition, No Starch Press, 2015
2. Reema Thareja “Python Programming Using Problem Solving Approach” Oxford University Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015.
4. Guido van Rossum and the Python development team, Python Tutorial Release 3.7.0, September 02, 2018.
5. Wesley J. Chun, “Core Python Applications Programming”, 3rd Edition, Pearson Education, 2016

Assessment Structure:

PBL	CIA	SEA	CIA (50)			SEA Conduction: 100 M Reduced to: 50 M
Conduction	50	50	Theory	I IA	II IA	Project Assessed for 100 marks reduced to 50 Marks
				25	25	
				Average of 2 tests – 25 M		
			Practical	Weekly Assessment (Record/Project) – 10 Marks Lab IA test – 15 Marks		
					Total – 50 Marks	**Total – 50 Marks**

i) CIA: 50%

Theory - 2 IA tests - Each of 25 Marks	25 Marks
Practical **Weekly Assessment** - Lab record/Project – 10 Marks **Lab IA test –** 15 Marks	25 Marks
Total	**50 Marks**

ii) SEA : 50%

Project	Write up – 10 Marks Project report – 25 Marks Presentation & Demonstration - 50 Marks Viva-Voce – 15 Marks	100 Marks Reduced to 50 Marks
	Total	**50 Marks**

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Semester: III / IV		
COURSE: CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS		
Course Code: 22CIP146	**L:T:P:J: 1:0:0:0**	**CIA Marks:** 50
Credits:	**1**	**SEA Marks: 5 0**
Hours:	**15 hrs**	**SEA Duration: 2Hrs**

Course Learning Objectives: The students will be able to	
1	know the fundamental political codes, structure, procedures, powers, and duties of Indian government institutions, fundamental rights, directive principles, and the duties of citizens
2	know the Indian top civil service positions and the exams conducted by UPSC and SPSC for the same
3	Understand engineering ethics and their responsibilities; identify their individual roles and ethical responsibilities towards society.

MODULE 1: Introduction to Indian Constitution	**RBT**	**Hrs**
The Necessity of the Constitution, Introduction to Indian Constitution, The Making of the Constitution, Role of Constituent Assembly, Preamble and Salient features of the Constitution of India, Fundamental Rights and its Restriction and limitations in different complex situations, Directive Principles of State Policy, Fundamental Duties.	**1,2,3**	**3**
MODULE 2: System of Government, Central Government, State Government	**RBT**	**Hrs**
System of Government-Parliamentary System, Federal System. Central Government-Basic details, Powers and Functions of Union Executive. Parliament- LS and RS (Composition, Duration, Membership and Presiding officers of Parliament and their functions). Leaders in Parliament (Leader of the House and Leader of the Opposition). Sessions of Parliament (Summoning, Adjournment, Adjournment Sine Die, Prorogation, Dissolution). Quorum of House, Language in Parliament, Joint sitting of two Houses. State Government-Basic details, Powers and Functions of State Executive. State Legislature (Composition, Duration, Membership and Presiding officers of Parliament and their functions).	**1,2,3**	**3**
MODULE 3: Judiciary, Amendments and Emergency Provisions	**RBT**	**Hrs**
Supreme Court, High Court, Judicial Review, Judicial Activism. Methods in Constitutional Amendments (How and Why). Types of Emergencies and its Consequences, Recent Amendments to the Constitution.	**1,2,3**	**3**
MODULE 4: Elections, Constitutional and Non Constitutional Bodies	**RBT**	**Hrs**
Elections- Election Commission of India, Electoral Process. Constitutional Bodies- Election Commission, Union Public Service Commission, State Public Service Commission, Goods and Service Tax Council. Non Constitutional Bodies- Central Information Commission, State Information Commission.	**1,2,3**	**3**

MODULE 5: Professional Ethics	**RBT**	**Hrs**
Scope & Aims of Engineering & Professional Ethics, Positive and Negative Faces of Engineering Ethics, Responsibilities in Engineering, the impediments to Responsibility. Trust and Reliability in Engineering, Risks, Safety and liability in Engineering, Clash of Ethics, IPRs (Intellectual Property Rights)	**1,2,3**	**3**

Course outcome: On completion of this course, students will be able to,
CO1: Have constitutional knowledge and legal literacy.
CO2: Have knowledge on All India Services and State Civil Services.
CO3: Understand Engineering and Professional Ethics and responsibilities of Engineers.

Reference Books

Suggested Learning Resources:

1.**Title of the Book - Indian Polity**
Name of the Author - M Lakshmikanth
Name of the Publisher-Mc Graw Hill Education
Edition and Year- 2019

2.**Title of the Book - Engineering Ethics**
Name of the Authors - M. Govindarajan, S.Natarajan, V.S. Senthilkumar
Name of the Publisher- Prentice-Hall
Edition and Year-2004

3. Durga Das Basu (DD Basu): “Introduction to the Constitution on India”, (Students Edition.) Prentice –Hall EEE, 19th / 20th Edn., (Latest Edition) or 2008.

4. Shubham Singles, Charles E. Haries, and Et al : “Constitution of India and Professional Ethics” byCengage Learning India Private Limited, Latest Edition – 2018.

5. M.Govindarajan, S.Natarajan, V.S.Senthilkumar, “Engineering Ethics”, Prentice –Hall of IndiaPvt. Ltd. New Delhi, 2004

6. M.V.Pylee, “An Introduction to Constitution of India”, Vikas Publishing, 2002.

7. Latest Publications of **NHRC - Indian Institute of Human Rights**, New Delhi.

Web Links and Video Lectures

www.unacademy.com/lesson/future-perfect-tense/YQ9NSNQZ https://successesacademy

Question paper pattern for SEA and CIA.

- The SEA question paper will be set for 50 marks and the pattern of the question paper will be objective type (MCQ).
- The CIA question paper will be set for 50 marks and the pattern of the question paper will be objective type (MCQ).

Final Marks = CIA + SEA = 50+50 = 100 Marks

Class Internal Assessment

IA1	Objective type questions 50Marks	Average of 2 IA will be taken 50Marks
IA2	Objective type questions 50Marks	
	Total CIA	**50 Marks**

Semester End Assessment

Semester end Exam	Objective type questions 50Marks	50 Marks
	Total SEA	**50 Marks**

Final Marks = CIA + SEA = 50+50 = 100 Marks