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: 23MAC131 (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

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	23AML132	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 organization of a computer system and the concept of programs as sequences of machine instructions.
- Describe arithmetic and logical operations with integer operands.
- Describe memory hierarchy and the hardware requirements for cache memory.
- Demonstrate different ways of communicating with I/O devices and standard I/O interfaces.
- 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.	**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.	**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**

Module 4 : INPUT - OUTPUT ORGANIZATION		
Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Interface Circuits	**8**	**(CO4) Apply**
Module 5 : BASIC PROCESSING UNIT		
Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization, Hard-wired Control **Pipelining:** Basic concepts of pipelining, parallel processing	**8**	**(CO5) Apply**

Course outcomes:

The students will be able to

1. Identify the basic organization of a computer system and the concept of programs as sequences of machine instructions. (Apply)
2. Solve the various arithmetic operation in computer hardware (Apply)
3. Identify the memory hierarchy and the hardware requirements for cache memory. (Apply)
4. Identify different ways of communicating with I/O devices and standard I/O interfaces. (Apply)
5. Estimate the processor time and CPU usage. (Apply)
6. 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.

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 : 4

Course Code	23AML133	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

- 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 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 **Constraint Satisfaction Problems:** Crypt-arithmetic problem **Practical:** 1. Implement AO* Search algorithm. 2. Write a Program to Implement Breadth First Search using Python.	**6+4**	**(CO2) Apply**

Module-3 : Game Playing		
Adversarial Search: Nim Game problem, minimax procedure, alpha-beta pruning. **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 **Practical:** 1. Write a program to implement Missionaries and Cannibals. 2. Write a Program to Implement Tower of Hanoi	**6+4**	**(CO3) 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, forward and backward chaining. **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**	**(CO4) 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**	**(CO5) 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. Apply algorithms that can learn to play games and make decisions
4. Develop knowledge base sentences using propositional logic and first order logic for logical reasoning.
5. 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

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	23AML134	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:

- Develop the skills needed to efficiently manage and manipulate these data structures in various applications.
- Distinguish the conceptual and applicative differences in binary trees, binary search trees for effective data retrieval and manipulation.
- Apply heap trees, hash tables, and AVL trees to manage data efficiently.
- Apply graph theory concepts and algorithms to effectively solve graphical problems.

	Number of Hours	**Bloom's Level**
Module 1: Stacks and Queues		
Concept Learning: Introduction to structure and unions, 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	**6+4**	**CO1 Apply**

QUEUE • Display the status of Circular QUEUE Support the program with appropriate functions for each of the above operations.		
Module 2: Linked List		
Linked Lists: Definition, Create, Insert, Delete, Update, Traverse, and Position-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). • Create a DLL of N 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	**6+4**	**CO2** **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 of complete binary tree. 2. Write a C program to print all the path from root to leaf 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**	**CO3** **Apply**
Module 4: Advanced Trees & Hashing		
Concept Learning: Heap Tree: Definition, Implementation, Insert, Delete,	**6+4**	**(CO4)**

Peek operations. Hashing: Hash Table, Hash Functions, Collision Handling by Open Addressing, Chaining, AVL tree. **Practical:** 1. Write a C program to construct MAX-Heap. 2. Write a C program to implement hashing technique and collision resolution technique.		**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 operations on Graph(G) of Cities, create a Graph of N cities using adjacency Matrix and print all the nodes reachable from a given starting node in a digraph using DFS/BFS. 2. Write a C Program to detect Cycle in a Directed Graph 3. Write a C Program to find if there is a path between two vertices in a directed graph.	**6+4**	**(CO5) Apply**
Course outcomes: The students will able to 1. Apply and implement fundamental data structures like stacks, queues, and pattern matching algorithms. 2. Apply and manipulate various types of linked lists, including practical applications such as polynomial addition. 3. Demonstrate the ability to use tree structures for different traversal methods and implement binary search trees for data retrieval and manipulation. 4. Apply heap trees, hash tables, and AVL trees for efficient data management. 5. Apply graph theory concepts and algorithms to solve graphical problems effectively.		
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		

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 : 3

Course Code	23AML135	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

- 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: $Y = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + ABCD$ 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.

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	23AML136	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 student USN, name, branch and semester and display information.	**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 Car that inherits from base class Vehicle using private strings and getter/setter methods to achieve encapsulation. 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 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. 5. Write a java program to create an abstract class named Shape that	**2**	**(CO1) Apply**

contains two integers and an empty method named print Area().		
Task 3		
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 4		
Exception Handling 1. 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. 2. 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 5		
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 thathas 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 6		

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 of a 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 7		
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 elements and Display elements. 5. Write a program to add elements to the HashMap given the key and value data type is string, get size of HashMap, and check if HashMap is empty.	**2**	**(CO1) Apply**
Task 8		
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**	**(CO3) Apply**
Task 9		
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 is fired.	**2**	**(CO3) Apply**
Task 10		

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**	**(CO3) Apply**
Task 11		
Event Handling 1. Write a java program to create a frame that contains two buttons and 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. When a user selects a particular item from the drop-down then it is shown on the top of the menu.	**2**	**(CO3) 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' as parameter.	**2**	**(CO1) 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**	**(CO2) Apply**

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.		
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 limited to 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, 7th 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.

BNM Institute of Technology

Syllabus for Softskills-1
SEMESTER – III

Subject Name	Softskills-1 (Soft skills for Industry & Quantitative Aptitude and Logical Reasoning)	**Weekly Continuous Assessment Marks (6 tests)**	Max 10 Min 4
Subject Code	23SFT	**Practice Tests for Internal Assessment Marks (6 tests)**	Max 15 Min 6
Number of Contact Hours/Week	2	**Final Assessment Marks (during 3rd test on Tab MCQ)**	Max 25 Min 10
Total Number of Contact Hours	24	**Credits**	1

1.Understanding Corporate Communication: (2hrs)	Define corporate communication and its importance. Identify key stakeholders in corporate. Understand workplace etiquette: communication tone, addressing colleagues, and handling conflicts. Resume Writing.
2.Cross Culture Sensitivity: (2hrs)	Knowledge, awareness and acceptance of other cultures/caste/creed. Being of oneness and understanding the work environment.
3. Effective Communication and Interpersonal Skills (2hrs)	Skills we use daily when we communicate and interact with other people, both individually and in groups. Active listening, right response and clear message. Conduct mock presentations on technical topics relevant to engineering.
4. Collaboration and Team Communication: (2hrs)	Participate in team-based activities and projects to practice collaboration. Understanding types of Leadership and inculcating leadership skills
5. Written Communication (2hrs)	Passage / Email writing Identify your goal. Before you write an email, ask yourself what you want the recipient to do after they've read it. Writing technical documentation and reviews
6. Goal Setting & Mind Mapping (2hrs)	Development of an action plan designed to motivate and guide a person or group toward a goal. Gives one's life direction and boosts your motivation and self-confidence. Developing career Action plans
7. Quantitative Aptitude - 1(6hrs)	**Speed Math** : Number System, Multiple and factors, Divisibility Rules HCF & LCM, Squares and Cubes, BODMAS & Tables, Approximation, Decimals, Fractions, Surds & Indices
	Profit & Loss - Concepts of SP, CP, Profit, Loss, Gain or Loss %. Marked Price & Discount problems, Successive Selling Tye, Discount
	Percentages – Percent to Decimal or Fraction Conversion, Inverse Case – Value From Percentage, relative Percentage, Successive Selling type, Dishonest Dealings, partnerships **Averages** - Understanding Averages, relevance, meaning, properties of average, deviation method, weighted average, & solving problems.
8. Logical Reasoning - 1 (6hrs)	**Logical Aptitude** – Syllogism, Venn diagram method, three statement syllogism, deductive and inductive reasoning, introduction to puzzle and games organizing information, parts of an argument, common flaws, arguments and assumptions.

Coding & Decoding – Letter Coding, Number coding, symbol coding, **Crypt arithmetic** – basic concepts, addition, subtraction, multiplication of coded alphabets, types of cryptarithmetic Concept of EJOTY, Opposite Letter, Reversing thealphabets. Jumbling of Letter, Finding Codes of Derivatives.
Image Analysis - Paper cutting & Folding, Mirror & Water Image, Cubes andDice, Analogy, Find the odd one out, Rule Detection. Cubes and dice
Series Completion - Basics of Next no, Missing no and Wrong no and problemson that. Solving various types of Letter series and understanding different types.
Odd Man Out - Following certain patterns and groups. Identifying the errors/odd one in the group.
Visual sequence, visual analogy and classification, single & multiple comparisons, linear sequencing
Logical Puzzles - K-level thinking, Arithmetic Puzzles and stick puzzles

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: 23MAC141 (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/

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: **23AML142**	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 basic concepts of OS, Process management and CPU scheduling to improve CPU utilization
- Illustrate process synchronization and Deadlocks in managing concurrent processes.
- Illustrate the concepts of main memory, virtual memory and secondary memory such as SSD for effective memory management.
- Introduction to Unix File System and file related commands
- Illustrate file attributes and shell programming to write simple shell scripts

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 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, Basic and extended regular expressions, 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	8	**(CO5) Analyze**

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

1. Apply the concepts of process management and process scheduling techniques to enhance CPU efficiency and throughput.
2. Apply process synchronization mechanisms, Identify the need of policies and protection required in managing deadlock
3. Apply memory management concepts to optimize memory usage and performance.
4. Apply the concepts of Unix system and file commands to perform various tasks in files and system.
5. Analyze the concepts of file attributes and manage files through shell programming to automate tasks

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/

BNM Institute of Technology

Dept. of Artificial Intelligence & Machine Learning

Choice Based Credit System (CBCS) and Outcome Based Education (OBE)

Semester: IV

Course Name: Database Management System		**Course Code: 23AML143**
L: T: P: J	3: 0 :1 :1	**CIA Marks:** 50
Credits:	4	**SEA Marks:** 50
Hours/Week (Total)	5	**SEA Duration:** 03 Hours

Course Learning Objectives: The students will be able to

1	Understand fundamental concepts, terminology and application of databases, SQL and NoSQL
2	Design concepts and creation of relational databases using relation algebra.
3	Practice SQL programming through a variety of database problems.
4	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	8+2	**Understand CO1**
Module-2: Relational Data Model and Relational Algebra		
Concept of relations, schema-instance distinction, keys, referential integrity and foreign keys, converting the database specification in E/R notation to the relational schema. relational algebra operators: selection, projection, set operators, cross product, various types of joins, division, example queries, Basic SQL: SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL. **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) 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)	8+2	**Apply CO2**

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. 3. Delete a book in BOOK table. Update the contents of other tables toreflect this data manipulation operation.		
Module-3: 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.) 4. Create a view that finds the salesman who has the customer with the highest order.	8+2	**Apply CO3**
Module-4: Functional Dependencies and Normalization		
Basics of Functional Dependencies and Normalization for Relational Database: Informal design guidelines for relation schema, Functional Dependencies, Armstrong's axioms for FD's, closure of a set of FDs, Equivalence of set FD's, minimal covers, Normal forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce- Codd Normal Forms [BCNF], Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form. Decomposition and Dependency Preservation, Decomposition and Lossless (Nonadditive) Joins. **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) 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	8+2	**Analyze CO4**

department that controls the project. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. **3.** 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.	**8+2**	**Analyze CO5**

Course Outcomes: After completing the course, the students will be able to	
CO1	Understand the Database System Concepts along with Data Modeling Using the Entity-Relationship (ER) Model
CO2	Apply the concepts of relations on RDBMS, constraints, joints using relational algebra operators.
CO3	Apply Structured Query Language for database manipulation.
CO4	Analyze functional dependencies to normalize relations of relational database
CO5	Analyze transactions processing, schedules protocols, serializability issues, deadlocks in DBMS and concepts of NoSQL with its advantages

Text Books

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

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	23AML144	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

- Understand and apply mathematical relations to analyze the asymptotic runtime complexity of algorithms.
- Utilize brute force, divide and conquer, and decrease and conquer methodologies to solve various computational problems.
- Synthesize efficient algorithms in common engineering design situations.
- Solve problems using backtracking and branch and bound methods, and gaining insights into the theoretical limits of algorithmic problem-solving and computational complexity.

	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. Apply brute force technique to recursively implement the following a. Linear Search b. To find the maximum and minimum from a given list of n elements. 2. 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. 3. Find the next three terms of the sequence 15, 23, 38, 61, … Fibonacci series of the given number using recursion.	**6+4**	**(CO1) Analyze**
Module-2: Brute Force, Divide and conquer		
Brute Force: Selection Sort **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. Write a program to search a key in a given set of elements using Binary search method and find the time complexity required to find the key. 3. 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 versustime taken. Specify the time efficiency class of this algorithm. 4. 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 versustime taken. Specify the time efficiency class of this algorithm. 5. Implement Topological sort using source removal method find the time required to sort the elements.	**6+4**	**(CO2) 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. Write a program to find maximum profit using Knapsack technique. 2. Implement Prim's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 3. Implement Kruskal's algorithm and Find Minimum Cost Spanning Tree of a given connected undirected graph. 4. Implement Dijkstra's algorithm find shortest paths to other vertices from a given vertex in a weighted connected graph.	**6+4**	**(CO3) Apply**
Module-4: Dynamic Programming		
The General Method with examples, Multistage Graphs. Transitive Closure: Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, Single-Source Shortest Paths: General Weights, 0/1 Knapsack, The Traveling Salesperson problem. **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. Implementation of Travelling Sales Man Problem using dynamic programming.	**6+4**	**(CO4) Apply**
Module-5: Backtracking		

Backtracking: General method (T2:7.1), Backtracking: n - Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem, Graph coloring, **Branch-and-Bound:** Assignment Problem, Knapsack Problem, Traveling Salesperson Problem. Basic concepts of P, NP, NP-Complete and NP-Hard Problems, Challenges of Numerical Algorithms. **Practical:** **1.** Implementation of N Queen Problem using Backtracking technique. **2.** Implementation of SUM-SUBSET Problem. **3.** Design and implement the presence of Hamiltonian Cycle in an undirected Graph G of n vertices. **4.** Implementation of Knapsack Problem using Branch and Bound.	**6+4**	**(CO5) Apply**

Course outcomes:

The students will able to

1. Analyze the asymptotic runtime complexity of algorithms by using mathematical relations that helps to identify them in specific instances.
2. Apply and solve problems using brute force, divide and conquer techniques, decrease and conquer.
3. Apply problem-solving methodologies such as greedy method to solve a given problem.
4. Apply dynamic programming to estimate the computational complexity of different algorithms.
5. Apply and solve problems using backtracking, branch and bound methods and describe P, NP and NP Complete problems.

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.

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	23AML145	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

- Understand the basic concepts of learning, machine learning and its landscape,
- Understand the concept of linear models for classification and regression,
- Understand the concepts of decision trees and support vector machines,
- Understand the concept of ensemble learning, and
- Understand Bayesian techniques for problem solving.

	Number of Hours	**Bloom's Level**
Module-1 : Introduction to Machine Learning		
Concept Learning: Concept Learning Task, Concept Learning as a Search, Maximally Specific Hypothesis, List-Then-Eliminate Algorithm, Candidate Elimination Algorithm, Inductive Bias **Machine Learning Landscape:** Types of Machine Learning (ML), Main Challenges of ML, Testing and Validation **Typical End-to-End Machine Learning Project:** Working with Real Data, The Big Picture, Getting Data, Discovering and Visualizing the Data, Data Preparation, Selecting and Training Models, Fine Tuning Model. **Practical:** 1. Perform data preprocessing and transformation considering an appropriate dataset.	**6+4**	**(CO1) Apply**
Module-2 : Linear Models for Classification & Regression		
Linear Models for Classification: Binary Classifier, Measuring Performance, Multiclass Classification, Multilabel Binary and Multilabel Multiclass Classification (Multioutput Classification), Error Analysis, Logistic Regression **Linear Models for Regression:** Linear Regression, Gradient Descent, Polynomial Regression, Learning Curves, Regularized Linear Models (Ridge Regression, Lasso Regression, Elastic Net Regression), Early Stopping	**6+4**	**(CO2) Apply**

Practical: 1. Build a machine learning model to classify a flower species using a linear model. 2. Build a machine learning model to predict house prices using linear regression.		
Module-3 : Decision Trees & Support Vector Machines		
Decision Trees: Training and Visualizing Decision Tree, Making Predictions, Estimating Class Probabilities, ID3 and CART Algorithm, Computational Complexity, Impurity Measurement – Gini and Entropy, Regularization, Regression, Issues in Decision Trees – Sensitivity to Axis Orientation and High Variance. **Support Vector Machines**: Linear and Non-linear Support Vector Machine (SVM) Classification, SVM Regression, Training Linear SVM Classifiers **Practical:** 1. Consider using Iris dataset to classify a flower using decision tree. 2. Continue with the same dataset to perform the same classification task with a model built using support vector machine algorithm.	**6+4**	**(CO3) Apply**
Module-4 Ensemble Learning		
Voting Classifiers, Bootstrap Aggregating (Bagging), Pasting, Out-of-bag Evaluation, Random Patches, Random Subspaces, Random Forests, Boosting – Adaptive and Gradient Boosting, Stacking **Practical:** **1.** Load an appropriate dataset and split it into a training set, a validation set, and a test set. Then train various classifiers, such as a random forest classifier, an SVM classifier, etc. Next, try to combine them into an ensemble that outperforms each individual classifier on the validation set, using soft or hard voting. Then try it on the test set and find if ensembling performs better compared to the individual classifiers.	**6+4**	**(CO4) Apply**
Module-5: Bayesian Learning		
Bayes Theorem and Concept Learning, Maximum Likelihood Hypothesis and Least-Squared Error Hypotheses, Minimum Description Length Principle, Bayes Optimal Classifier, Gibbs Algorithm, Naive Bayes Classifier, Bayesian Belief Network, Expectation-Maximization (EM) Algorithm **Practical:** 1. Draw a Bayesian belief network that represents the conditional independence assumptions of the naive Bayes classifier for an appropriate problem. Give the conditional probability table(s)	**6+4**	**(CO5) Apply**

associated with specified node(s).		

Course outcomes:

The students will able to

1. Apply the basic concepts learning and basic machine learning theory in machine learning problems.
2. Apply linear models for classification and regression tasks.
3. Apply decision trees and support vector machines in appropriate machine learning problems.
4. Apply ensemble learning to achieve relatively better model performance than individual ones.
5. Apply Bayesian techniques in an appropriate machine learning problem.

Reference Books:

1. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Third Edition, O'Reilly, 2022
2. Tom M. Mitchell. *Machine Learning*, McGraw Hill Education, Indian Edition, 2013

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – IV

Cloud Computing & Applications (PBL)

Credit: 2

Course Code	23AML146	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

- Acquire knowledge and be exposed to tool kits for cloud Environment.
- Understand the representation and familiar with developing web services / Applications in cloud environment.
- Develop the ability to run Virtual machines of different Configuration
- Develop the ability to implement and use parallel programming using Hadoop.

	Number of Hours	Bloom's Level
Task – 1		
Fundamentals of Cloud Computing: Overview of cloud computing concepts - Types of cloud services: IaaS, PaaS, SaaS - Advantages and disadvantages of cloud computing - Types of cloud services: IaaS, PaaS, SaaS and basic cloud security - Public, private, and hybrid cloud - Community cloud - Multi-cloud and Inter-cloud strategies	**2**	**Apply**
Task – 2		
AWS Fundamentals- Overview of Amazon Web Services (AWS) - Setting up an AWS account - Overview of AWS services (EC2, S3, RDS, VPC, IAM) **AWS Databases**- EC2 (Elastic Compute Cloud) for virtual servers - S3 (Simple Storage Service) for object storage - EBS (Elastic Block Store) for block storage - RDS (Relational Database Services)	**2**	**Apply**
Task – 3		
Understanding the GIT Repository, creating a GIT repository and executing the control system commands to Clone, Commit, Push, Fetch, Pull, Checkout, Reset and Delete	**2**	**Apply**
Task – 4		
Setting up a VIRTUALBOX/VMware Workstation, install a C Compiler in the virtual machine and execute a sample program	**2**	**Apply**
Task – 5		
Install GOOGLE APP ENGINE and Host A Static Website On GOOGLE APP ENGINE	**2**	**Apply**
Task – 6		
Simulate A Cloud Scenario Using Cloudsim And Run A Scheduling Algorithm ThatIs Not Present In Cloudsim	**2**	**Apply**
Task – 7		
Moving Files Between Virtual Machines.	**2**	**Apply**
Task – 8		
Introduction to Azure- Overview of Microsoft Azure - Setting up an Azure account - Overview of Azure services (Virtual Machines, Blob Storage, Azure SQL Database **Azure Virtual Machines -** Azure App Service - Azure Functions - Azure Virtual Network - Azure Security Center - Azure Active Directory	**2**	**Apply**

Task – 9		
Creating an azure account and implement data load	**2**	**Apply**
Task – 10		
Implementation of database operations like insert, delete, update using azure data factory.	**2**	**Apply**
Mini Project		

Course outcomes:

The students will be able to

1. Configure various virtualization tools such as Virtual Box, VMware workstation.
2. Design and deploy a web application in a PaaS environment.
3. Learn how to simulate a cloud environment to implement new schedulers.
4. Install and use a generic cloud environment that can be used as a private cloud.
5. Manipulate large data sets in a parallel environment.

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Semester: III / IV		
COURSE: CONSTITUTION OF INDIA AND PROFESSIONAL ETHICS		
Course Code: 23CIP147	**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

BNM Institute of Technology

Syllabus for Softskills-2
SEMESTER – IV

Subject Name	Softskills-2 (Quantitative Aptitude & Logical Reasoning)	**Weekly Assessment Marks (6 tests)**	Max 10 Min 4
Subject Code	23SFT147 /148	**Practice Tests for Internal Assessment Marks (6 tests)**	Max 15 Min 6
Number of Contact Hours/Week	2	**Final Assessment Marks (during 3rd test on Tab MCQ)**	Max 25 Min 10
Total Number of Contact Hours	24	**Credits**	1

Module 1 (Quantitative Aptitude - 2) (6hrs)	**Ratio and Proportion** · Simple Ratios, Compound Ratios · Comprehend and Dividend · Direct & Indirect Proportions · Problems on ages,
	Mixtures & Alligation– technique, general representation, the straight-line approach, application of weighted allegation methods in problems involving mixtures, application of allegation on situations other than mixtures problems & Coin problems.
	Data Interpretation – Simple arithmetic, rules for comparing fractions, calculating(approximation)fractions, shortcut ways to find the percentages, Classification of data tables, Bar Graph, Tabular Form, Line Chart, case let Form. Pie Chart, Radar/Web, combination of graphs, combination of graphs and tables. and Missing Data Interpretation.
	Simple interest and Compound Interest - Simple Interest, Basic Difference b/w both the Interests. Mixed interest, CI with a Fractional Rate, to find Installments of both SI and CI.
Module 2 (Logical Reasoning -	**Ages & Blood Relation** - Generation Tree, Family Tree Problems, Statement Based Questions, Coded Blood Relation Question, Blood Relation Based Puzzles.
	Seating Arrangement - Single or Double rows facing each other or away from each other in the same direction Circular Seating Arrangement · Uni- & Bi-directional problems on · Circular, Square, Rectangular, Hexagonal tables

2) (6hrs)	**Distance & Direction** - Distance and Displacement between any two points as well as puzzles based on that, Concept of Shadows.
Module 3 (Quantitative aptitude - 3) (6hrs)	**Speed, Time & Distance** - Relative speed, Average speed, Problems on Races when somebody gives a start to other Person. Persons moving opposite & same direction in circular track, problems on train, Boat & Stream,
	Permutation & Combination - Arrangement, law of addition & multiplication, factorial function, concept of step arrangement, permutation of similar things, circular things, find the rank of word in dictionary. Concepts of selections using combination.
	Time & Work - Partial work done problems Work done on Alternate days. Problems related to efficiency. Division of Wages According to Work. Work & wages, pipes & cisterns
Module 4 (Logical Reasoning - 3) (6hrs)	**Clocks & Calendars** - Understanding concepts and basic formula along with solving different types of problems. **Probability** - Single event probability, multi event probability, independent events and dependent events, mutually exclusive events, non-mutually exclusive events, combination method for finding the outcomes.
	Analogies - Drawing similarity between different but sufficiently similar events, situations, or circumstances using tips and tricks, find the odd one out, rule detection,

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – V

Software Engineering, Project Management and Finance (PCC)
Credit: 3

Course Code	23AML151	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
- Identify ethical and professional issues and explain why they are of concern to software engineers.
- Recognize the importance of software maintenance and describe the intricacies involved in software evolution.
- Apply estimation techniques, schedule project activities and compute pricing.
- Identify software quality parameters and quantify software using measurements and metrics.
- Recognize the need for agile software development, describe agile methods, apply agile practices and plan for agility.

Prerequisites:
Basic knowledge of programming, math, logic, software evolution, quality, and ethical practices in software engineering.

	Number of Hours	**Bloom's Level**
Module-1: Introduction		
Introduction: Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies. **Software Processes:** Models: Waterfall Model, Incremental Model and Spiral Model, Process activities. **Requirements Engineering:** Requirements Engineering Processes, Functional and non-functional requirements. **Self-Study:** The software Requirements Document, Requirements Specification, Requirements validation, Requirements Management	**8**	**(CO1) Apply**
Module-2: System Models, Design and Implementation, Software Testing		
System Models: Structural models, Behavioral models, UML modeling using StarUml tool. **Design and Implementation**: Introduction to RUP, Design Principles **Software Testing:** Development testing, Test-driven development. **Self Study:** Release testing, User testing.	**8**	**(CO2) Apply**
Module-3: Project management, Project Planning, Quality management		
Project management: Risk management, Managing People, Teamwork. **Project Planning:** Software pricing, Plan-driven development, Project scheduling: Estimation techniques,	**8**	**(CO3) Apply**

Quality management: Software quality, Reviews and inspections **Self Study:** Software measurement and metrics, Software standards		
Module-4: Agile Software Development		
Agile Software Development: Coping with Change, The Agile Manifesto: Values and Principles. Agile methods: SCRUM (Ref ―The SCRUM Primer, Ver 20.) and Extreme Programming. Plan-driven and agile development. **Self Study :** Agile project management, Scaling agile methods.	**8**	**(CO4) Apply**
Module-5: Managing Project Finances		
How to Manage Project Finances-Cost estimating: Work Breakdown Structure, Cost budgeting: Cost Aggregation, Reserve Analysis, Parametric estimating, Infrastructure and overheads, Cost control: Change Control, Resource Management **Performance Measurement and Analysis-** Cost Variance, Earned Value, Schedule Variance, Cost Performance Index, Schedule Performance Index**.** **Self Study:** Forecasting, Introduction of Tools to manage project Finances- TouchBase Project Financials	**8**	**(CO5) Apply**
Course outcomes: 1. Understand the activities involved in software engineering and identify the role of various process models. 2. Design a software system, component, or process to meet desired needs within realistic constraints and describe various software testing methods 3. Illustrate the role of project planning and quality management in software development. 4. Describe agile project management and benefits of using agile approaches. 5. Understanding financial concepts and apply them to control Project Costs.		
Reference Books: **1.** Software Engineering Ian Sommerville Pearson Education 10th Edition, 2012 **2.** Software Engineering Ian Sommerville Pearson Education 9th Edition, 2012 **3.** Software Engineering-A Practitioner approach Roger S. Pressman Tata McGraw Hil 9th Edition **4.** An Integrated Approach to Software Engineering Pankaj Jalote Wiley India **5.** A guide to the project Management body of knowledge- PMBOK guide , 7th edition		

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – V

Automata Theory and Computations (PCC)
Credit: 3

Course Code	23AML152	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	2:1:1:0	SEA Marks	50
Total Number of Lecture Hours	40	Exam Hours	03

Prerequisite Concepts: Data Structures, Algorithms and Complexity, Discrete Mathematics

Course Learning Objectives:

This course will enable students to

- Introduce core concepts in automata and theory of computation.
- Identify different formal language classes and their relationships.
- Design grammars and recognizers for different formal languages.
- Prove or disprove theorems in automata theorem using their properties.
- Determine the decidability and intractability of Computational problems.
- Design and develop lexical analyzers and parsers.

	Number of Hours	**Bloom's Level**
Module-1: Introduction		
Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, Finite State Machines (FSM): Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, Minimizing FSMs, Canonical form of Regular languages, Finite State Transducers, Bidirectional Transducers. **Self-Study Components:** ε- NFA. **Use Case:** Construct different kinds of FSM using JFLAP • Python program to construct a DFA which accept the language L = {anbm | n mod 2=0, m≥1} • Python program to construct nfa for the language nfa (a|b)*abb	**8**	**(CO1) Apply**
Module-2: Regular Expressions & Languages		
Regular Expressions (RE): Introduction to RE, Kleene's theorem, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non regular Languages: How many RLs, To show that a language is regular, to show some languages are not RLs. **Self-Study Components:** Applications of REs, Closure properties of RLs **Use Case:** Develop regular grammars and check closure properties using JFLAP • Python program for matching a Regular Expression. • Python program for Regular Grammar.	**8**	**(CO1) Apply**

Module-3: Context-Free Grammars & Pushdown Automata		
Context-Free Grammars(CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. **Pushdown Automata (PDA)**: Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs. **Self-Study Components:** Nondeterminism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA. **Use Case:** Construct PDA for the context free grammar using using JFLAP • Derive a parse tree using python programming • Construct a PDA using python programming for a set of languages	**8**	**(CO2) Apply**
Module-4: Turing Machine		
Turing Machine: Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. Variants of Turing Machines (TM), The model of Linear Bounded automata. **Decidability:** Definition of an algorithm, decidability, decidable languages, Undecidable languages, halting problem of TM, Post correspondence problem. **Self-Study Components:** Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis**.** Applications: G.1 Defining syntax of programming language. **Use Case:** Construct Turing machine and check decidability and undecidability of languages using JFLAP • Construct a DTM which matches all strings beginning with '0's, and followed by the same number of '1's • Turing machine for the language L = {a^mb^na^mb^n ∣ m,n≥0}	**8**	**(CO3) Apply**
Module-5: Lexical Analysis & Syntax Analysis		
Lexical Analysis: role of lexical analyzer, input buffering, specification of tokens, recognition of tokens, design of lexical analyzer generator. **Syntax analysis**: role of parser, top down parsing, bottom up parsing, operator precedence parsing, LR parsers, parser generators. **Self-Study Components:** language for specifying lexical analyzers, from Regular expression to NFA, **Use Case:** Apply Automata concepts to check syntax of programming language. • Implement lexical analyzer using python	**8**	**(CO4) Apply**
Course outcomes: The students will be able to: 1. Apply the core concepts of Automata Theory and Computation and convert different automata models, Regular Expressions to FSM. (Apply) 2. Develop Grammars and Automata for different language classes. (Apply) 3. Apply the concept of Turing machine, decidability and undecidability on the grammar (Apply) 4. Design and develop lexical analyzers and parsers. (Apply) 5. Analyze different type for models using simulators (Analyze)		

Reference Books:

1. Elaine Rich, Automata, Computability and Complexity, 1st Edition, Pearson Education, 2018.
2. K L P Mishra, N Chandrasekaran, 3rd Edition, Theory of Computer Science, PHI, 2012.
3. John E Hopcroft, Rajeev Motwani, Jeffery D Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education, 2013
4. Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers-Principles, Techniques and Tools, Pearson, 2nd Edition, 2013.

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Artificial Intelligence and Machine Learning

Computer Networks and Security (PCI)
Credit: 4

SEMESTER– V

Course Code	23AML153	CIE Marks:	50
Teaching Hours/Week (L: T: P: J)	3:0:2:0	SEE Marks:	50
Total Number of Lecture Hours	50	SEE Duration:	3 Hours

Course Learning Objectives:

This course will enable students to

- Apply the networking components, models, and topologies to solve various real time problems.
- Describe and apply the physical layer, data link layer, network layer, transport layer and application layer architectures, protocols and services Provided by each layer for networking.
- Identify the security threats, vulnerabilities and services along with applying cryptographic solutions for it.
- Analyze a data transmission at different layer and parameter measures for real time network using simulation tools.

Prerequisite: Basics of coding, Mathematics

Module 1: Introduction	**Number of Hours**	**Bloom's Level**
Data communication: Components, Data representation, Data flow, Networks: Network criteria, Physical Structures, Network types: LAN, WAN, Switching, The Internet. TCP/IP PROTOCOL SUITE, Layered Architecture, Layers in the TCP/IP Protocol Suite, Description of Each Layer, Encapsulation and De-capsulation, Addressing, Multiplexing and De-multiplexing, OSI versus TCP/IP. **Physical Layer**: Data and Signals, Periodic analog signals, digital signals, Transmission impairment. **Practical Component using NS2:** 1. Implement three nodes point – to – point network with duplex links between them. 2. Implement the following topologies: Bus, Star, Ring Self-Study: Line coding schemes	**6+4**	**(CO1) Apply**
Module 2: Data Link Layer		

Data-Link Layer: Nodes and Links, Services, Two Categories of link, Sublayers, Link Layer addressing: Types of addresses, ARP. Data Link Control (DLC) services: Framing, Flow and Error Control, Data Link Layer Protocols: Simple Protocol, Stop and Wait protocol, Wired and Wireless LANs: Ethernet Protocol, Standard Ethernet. **Practical Component using NS2:** 1. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion. 2. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination. Self-Study: Error detection and correction: CRC	**6+4**	**(CO2) Apply**
Module 3: Network Layer		
Network Layer: Introduction, Network Layer services: Packetizing, Routing and Forwarding, Packet Switching: Datagram Approach, Virtual Circuit Approach. IPV4 Addresses: Address Space, Classful Addressing, Classless Addressing, DHCP, Network Address Resolution, Distance Vector Routing, Link State Routing, Path vector routing. **Practical Component using NS2:** 1. Simulation of distance vector routing algorithm. 2. Simulation of link state routing algorithm Self-Study: IPV6 Addressing	**6+4**	**(CO3) Apply**
Module 4: Transport and Application Layer		
Transport Layer: Introduction: Transport Layer Services, Connectionless and Connection-oriented Protocols, Transport Layer Protocols: Simple protocol, Stop and wait protocol, Go-Back-N Protocol, Transport-Layer Protocols in the Internet: User Datagram Protocol: User Datagram, UDP Services, Transmission Control Protocol: TCP Services, TCP Features. **Application Layer:** Introduction: providing services, Application- layer paradigms, Standard Client -Server Protocols: World wide web, Hyper Text Transfer Protocol, FTP: Domain Name system. **Practical Component using NS2:** 1. Implement Transport Control Protocol in sensor network 2. Implement User Datagram protocol in sensor network **3.** Simulation of stop and wait protocol and sliding window protocol **Cisco Packet Tracer:** 4. Configuration of TELNET protocols on router for remote access. Self-Study: Electronic Mail	**6+4**	**(CO3) Apply**
Module 5: Network Security		
Security Introduction, Attacks, Services, Mechanism, network security, Cryptography, DES, AES, Public-Key Cryptography: RSA, Hash and MAC Algorithms: Secure Hash Algorithm (SHA), Digital Signatures. **Practical Component using JAVA:** 1. Write a Java program to implement the DES algorithm logic 2. Write a Java program to implement RSA Algorithm 3. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.	**6+4**	**(CO4)**

4. Basic lab Program using POSTMAN API. Self-Study: Diffie-Hellman Key Exchange		**Apply**

Course Outcomes:

At the end of the course, the students will be able to:

1. Apply the concepts of networking and data transmission to solve various real time problems.
2. Apply the various services and protocols of data link layer.
3. Apply and analyze the various networking architectures, protocols and services of network and transport layers for real time networking.
4. Identify the networks security, threats, and vulnerabilities and Apply the different cryptographic operations.

Reference Books

1. Behrouz A Forouzan, "Data Communications and Networking", 5th Edition, McGraw Hill, 2013, ISBN: 1-25-906475-3.
2. Cryptography and Network Security Principles and Practice Fourth Edition, William Stallings, Pearson Education
3. James J Kurose, Keith W Ross, "Computer Networks", Pearson Education.
4. Andrew S Tanenbaum, 'Computer Networks", Prentice Hall.
5. Modern Cryptography: Theory and Practice, by Wenbo Mao, Prentice Hall PTR
6. Network Security Essentials: Applications and Standards, by William Stallings. Prentice Hall

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Artificial Intelligence and Machine Learning

Advanced Machine Learning (PCI)
Credit: 4

SEMESTER– V

Course Code	23AML154	CIE Marks:	50
Teaching Hours/Week (L: T: P: J)	3:0:2:0	SEE Marks:	50
Total Number of Lecture Hours	50	SEE Duration:	3 Hours

Course Learning Objectives:

This course will enable students to

- Understand concerns related to dimensionality and various approaches for efficient model training and data visualization,
- Understand unsupervised learning techniques, and perform clustering and anomaly detection,
- Understand the fundamentals of time-series and perform modeling to forecast over time-series data,
- Understand various approaches towards instance- or memory-based learning and build models, and
- Learn the concept and implementation of brain inspired artificial neural networks.

Prerequisites: Python, Machine Learning Basics

Module 1: Dimensionality Reduction	**Number of Hours**	**Bloom's Level**
Curse of Dimensionality, Main Approaches for Dimensionality Reduction – Projection and Manifold Learning, Principal Component Analysis (PCA), Random Projection, Locally Linear Embedding, Other Dimensionality Reduction Techniques – *t*-Distributed Stochastic Neighbor Embedding (t-SNE) **Practical:** 1. Apply PCA to reduce an appropriate dataset preserving a given explained variance. 2. Consider the same or an appropriate dataset to experiment with data decompression and incremental PCA. Self-Study: Linear Discriminant Analysis (LDA)	**6+4**	**Apply [CO1]**
Module 2: Clustering		
k-Means – The *k*-Means algorithm, Centroid initialization methods, Accelerated *k*-Means and mini-batch *k*-Means, Finding the optimal number of clusters; Limits of *k*-Means, Clustering for Semi-Supervised Learning; DB Scan; Other Clustering Algorithms Gaussian Mixtures – Gaussian Mixtures for Anomaly Detection, Selecting the Number of Clusters, Bayesian Gaussian Mixture Models **Practical:** 1. Develop a program to apply *k*-Means algorithm to cluster a set of instances on an appropriate dataset. 2. Develop a program to train a Gaussian mixture model on an appropriate dataset. Use the model to generate some new samples and visualize them.	**6+4**	**Apply [CO2]**

Self-Study: Anomaly and Novelty Detection, Other Algorithms for Association Rule		
Module 3: Time-series Forecasting		
Introduction to time-series forecasting, Components and decomposition, building baseline models with naive approaches; Random Walk – Stationarity, Autocorrelation function (ACF); Moving Average Process, Autoregressive Process- Autoregressive Moving Average (ARMA) model **Practical:** 1. Consider an appropriate time-series dataset and forecast using a base model and a moving average model. Self-Study: ARIMA and Seasonal ARIMA model	**6+4**	**Apply [CO3]**
Module 4: Instance-based Learning		
k-Nearest Neighbor Learning **-** Distance-Weighted Nearest Neighbor Algorithm and remarks; Locally Weighted Regression **-** Locally Weighted Linear Regression and remarks; Radial Basis Function; **Practical:** 1. Write a program to implement *k*-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. 2. Implement the non-parametric locally weighted regression algorithm to fit data points in an appropriate data set. Self-Study: Cased-based Reasoning; Remarks on Lazy and Eager Learning	**6+4**	**Apply [CO4]**
Module 5: Artificial Neural Networks		
From Biological to Artificial Neurons: Biological Neurons, Logical Computations with Neurons, The Perceptron, Multilayer Perceptron (MLP) and Backpropagation, Regression & Classification MLPs **Implementing MLPs with Keras**: Classification and Regression MLPs using the Sequential API, Complex Models using the Functional API, Callbacks **Practical:** 1. Use Scikit-Learn to build an MLP, train on appropriate dataset for a regression task, perform predictions and then measure performance. 2. Use Keras sequential API to build and train a multiclass image classifier model, perform classification, and then evaluate performance. Self-Study: Fine-Tuning Neural Network Hyperparameters, TensorBoard	**6+4**	**Apply [CO5]**
Course Outcomes: At the end of the course, the students will be able to:		

1. Apply dimensionality reduction techniques for efficient model training and data visualization
2. Apply clustering, and detect abnormal (anomaly) and unusual (novelty) observations,
3. Perform forecasting by implementing appropriate models over time-series data,
4. Implement appropriate instance-based learning algorithms to solve relevant prediction tasks, and
5. Implement regression and classification using MLPs.

Reference Books

1. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Third Edition, O'Reilly, 2022
2. Marco Peixeiro. *Time Series Forecasting in Python*, Manning, 2022
3. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 2013

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER - V

Virtual Reality and Augmented Reality(PBL)
Credit 2

Course Code	23AML155	CIA Marks	50
Teaching Hours/Week (L:T:P:J)	0:0:2:2	SEA Marks	50
Total Number of Hours	30	Exam Hours	03

Course Learning Objectives:

- Experience the fundamental Computer Vision, Computer Graphics and Human-Computer interaction Techniques related to VR/AR
- Demonstrate the Geometric Modelling Techniques Review the Virtual Environment
- Develop VR/AR Technologies Simulate and Apply Virtual/Augmented Reality to varieties of Applications.

	Number of Hours	Bloom's Level
Task – 1		
Blender Introduction • Demonstrate Blender Interface, Overlay Reference, Position on Overlay, Managing Layouts, Camera Overlay • Demonstrate Scene view navigation, Scene view camera, Game View. • Project Creation, Plane creation, cube creation, Explanation of cube camera and position, Material Selection, and class file same, adding motion to the script, Run. • Understanding different 3D projectiles	**4+ 2 (Project)**	**Apply**
Task – 2		
Blender Programming • Show the use of blender for Creating, loading and saving the scenes. • Demonstrate the Transforms, Components, 3D objects, creating components, Saving the work. • Demonstrate the Layers, Constraints, Editor Feature 3d mode, Preferences, Build settings, Sculpting, preference add Ons • Installation of XR device, Viewing through the XR. • Introduction to Unity	**4+2 (Project)**	**Apply**
Task – 3		
Graphics • Demonstrate the use of Render Pipelines, Cameras, lighting, models, mirroring objects • Demonstrate the use of Meshes, Textures, shaders, materials, Image Textures • Demonstrate the concept Visual effects, sky, colour for suitable example. • Demonstrate the concept of World Building, Terrain, tree Editor for suitable example.	**4+2 (Project)**	**Apply**

• Demonstrate XR Plug-in Management, Installation of packages. • Demonstrate Creation of left-hand, left-hand Controller and right-hand controller. • Setting up the unity and programming using Unity.		
Task 4		
Scripting, Audio Video and Animation • Demonstrate the Setting up scripting Environment. • Apply the concepts for Creating frames, creating .mkv files, namespaces, attributes • audio files, tracker Modules, Audio Group inspector. • Demonstrate for suitable example to create Animation, Rotation in animation, animation clips, Humanoid avatars, Animation. • Bone movement and animation • Using Unity development of Animations : This work is done using programming.	**4+2 (Project)**	**Apply**
Task - 5		
Augmented reality • Program to show augmented reality. • Program to show The Relationship Between Augmented Reality and Other Technologies-Media, Technologies. • Program to show Spectrum Between Real and Virtual Worlds, applications of augmented reality Augmented. C# Programming of pivot object in unity	**4+2 (Project)**	**Apply**
Mini project • Using VR exploring the human body level by level, including cell level. • Using VR describing how medicine and body cures the illness. • Touchless ATM Using Augmented Reality. • Augmented Reality Controlled Hologram. • Augmented Reality House devices. • Augmented Reality Agriculture Field. • Augmented Reality and Medical devices. • VR Game Development. • VR Application Development. • Development of AI controlled VR Device, that accurately work to keep the clarity very good and using technology keep the eye healthy. • Development of Cardboard VR device as activity. • Improvising the quality of the VR Device and determining the accuracy. • Development of Satellite with the capability of VR Camera and more features. • Ability to view the Solar system using the VR Device using compatible camera sent through the satellite. • Upgrading the ARVR device		

Course Outcomes:

The students will be able to

1. Demonstrate proficiency in handling Unity using C# in using the graphics to develop a model, Interface, Navigation, and scenes.
2. Apply the working of different Real-world models, Meshes, Textures
3. Apply the skills in developing Humanoid, and the basic models of Augmented Reality Develop the real time projects to solve complex problems and see the results in visual effects.

References:

1. Virtual Reality Technology, Second Edition, Gregory C. Burdea & Philippe Coiffet, John Wiley & Sons.
2. Allan Fowler-AR Game Development‖, 1st Edition, A press Publications, 2018, ISBN 978-1484236178
3. Allan Fowler- Beginning iOS AR Game Development Developing Augmented Reality Apps with Unity and C#, 1st Edition, Apress Publications, 2018, ISBN 978-1484236178
4. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011
5. https://docs.unity3d.com/2023.2/Documentation/Manual/ScriptingSection.html
6. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016
7. Understanding Virtual Reality: Interface, Application and Design, William R Sherman and Alan B Craig, (The Morgan Kaufmann Series in Computer Graphics)". Morgan Kaufmann Publishers, San Francisco, CA, 2002

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Deep Learning (PCI)
Credit: 3

Course Code	23AML161	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	2:0:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Learn the various training aspects of a neural network,
- Learn the need for convolutional neural network for visual tasks,
- Learn the working of recurrent and convolutional neural network for sequence processing,
- Learn approaches for natural language tasks using recurrent neural networks and attention mechanism, and
- Learn the working on autoencoders to learn latent representations of input data to be used for various tasks.

Prerequisites: Basics of Machine Learning, Linear Algebra

	Number of Hours	**Bloom's Level**
Module-1: Deep Neural Networks Training		
The Vanishing/Exploding Gradients Problems, Reusing Pretrained Layers, Faster Optimizers, Learning Rate Scheduling, Avoiding Overfitting Through Regularization **Practical:** 1. Build a neural network of appropriate depth on CIFAR10 dataset with He initialization, Swish activation function, Nadam optimizer and early stopping. Compare this model's learning curve on its convergence, performance and training time with another model with batch normalization layer added. 2. Apply transfer learning to reuse pretrained layers to experiment if it reduces training time and improves model performance with less data. Self-study: Python code to use and compare different optimizers, initialization functions and variants of the ReLU activation function.	**6+4**	**(CO1) Apply**
Module-2: Convolutional Neural Networks		

Architecture of the Visual Cortex, Convolutional Layers, Pooling Layers, Convolutional Neural Network (CNN) Architectures, Data Augmentation, Pretrained Models from Keras, Pretrained Models for Transfer Learning, Classification and Localization, Object Detection – Sliding CNN, Fully Convolutional Networks, YOLO, Object Tracking, Semantic Segmentation **Practical:** 1. Build CNN from scratch on *MNIST* dataset. 2. Build an image classifier using a pretrained *Xception* model on TensorFlow flower dataset. Self-study: Building a model using pretrained ResNet and VGG Net	**6+4**	**(CO2) Apply**
Module-3: Sequence Processing		
Recurrent Neurons and Layers – Memory Cells, Input and Output Sequences; Training RNNs, Forecasting a Time-series, Handling Long Sequences - Unstable Gradients Problem, Short-Term Memory Problem **Practical:** 1. Analyze a simple time-series dataset and perform univariate time-series forecasting first with a linear model and then with a simple RNN and compare model performance. Also experiment with deep RNNs to check if the model performance improves further. 2. Consider the same dataset and perform multivariate time-series forecasting and compare model performance with the one from the previous experiment. Self-study: Implementation of Time-LLM to forecast time series data	**6+4**	**(CO3) Apply**
Module-4: Text Processing		
Generating Text using Character RNN; Sentiment Analysis – Masking, Reusing Pretrained Embeddings and Language Models; Encoder–Decoder Network for Neural Machine Translation - Bidirectional RNNs, Beam Search, Attention Mechanisms - Transformer Architecture and models; Vision Transformers; Hugging Face's Transformers Library **Practical:** 1. Consider *IMDb* reviews dataset and train two models – one without any pretrained embeddings and other one with a contextualized pretrained embeddings, to classify sentiment of a movie review and then compare performance of these two models. 2. Use both pipeline and pretrained transformer language models from Hugging Face Transformer library to generate text from a prompt. Self-study: Bidirectional LSTM	**6+4**	**(CO4) Apply**
Module-5: Representation Learning using Autoencoders		

Efficient Data Representations, Performing PCA with an Undercomplete Linear Autoencoder, Stacked Autoencoders, Convolutional Autoencoders, Denoising Autoencoders, Sparse Autoencoders, Variational Autoencoders **Practical:** 1. Consider an appropriate image dataset and build denoising stacked autoencoder to reconstruct noisy images. 2. Build a variational autoencoder on the appropriate image dataset and use it to generate images. Self-study: Comparision of different types of autoencoders and their usage in a specific field.	**6+4**	**(CO5) Apply**

Course outcomes:

The students will be able to

1. Use pretrained layers, choose appropriate optimizer and learning rate, and avoid model overfitting
2. Design appropriate CNN architectures, apply transfer learning, use pretrained models, and perform various image tasks
3. Apply RNN and CNN in various sequence processing tasks
4. Apply pretrained embedding and transformer models for natural language tasks
5. Apply autoencoders and PCA in various unsupervised machine learning tasks.

Reference Books:

1. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, Third Edition, O'Reilly, 2022
2. Charu C. Aggarwal. *Neural Networks and Deep Learning*, Second Edition, Springer, 2023
3. François Chollet, Matthew Watson. *Deep Learning with Python*, Third Edition, 2025, Manning
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*, MIT Press, 2016
5. Michael Nielsen. *Neural Networks and Deep Learning*, 2019

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Natural Language Processing (PCI)
Credit: 3

Course Code	23AML162	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	2:0:2:0	SEA Marks	50
Total Number of Lecture Hours	50	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Understand Natural Language Processing Concepts and its Applications.
- Analysis of regular expressions, parsing.
- Semantic Analysis of meaning representation.
- Design of information retrieval models.
- Apply NLP concepts in language model.

Prerequisite Concepts: Automata Theory Concepts, Probability, Python Programming Language.

	Number of Hours	**Bloom's Level**
Module-1: Introduction & Word Level Analysis		
Introduction: Introduction to Natural Language Processing, Stages in natural language Processing, NLP Pipeline, Origins and challenges of NLP, Applications of NLP, Introduction to the corpus. **Word level Analysis:** Regular Expressions-Finite-State Automata-Morphological Parsing-Spelling Error Detection and correction. Part-of-Speech Tagging- Rule based tagger, Stochastic tagger, Hybrid Taggers, Unknown words. **Self-Study Component**: Corpus, Basic elements of corpus, Regular Expressions-Finite-State Automata. **Practical:** 1. a) Tokenizing -Design a Python program to splitting up a larger body of textinto smaller lines, words or even create words for a non-English language. 2. Lemmatizing & Stemming- Design a Python program to group together the different inflected forms of a word so they can be analyzed as a single item. 3. Corpus- Design a Python program to illustrate corpus. 4. Process-Implement a python program to process the given text. 5. a) Chunking- Design a python program to group similar words together based on the nature of the word. b) Chinking- Design a Python program to remove a sequence of tokens froma chunk. 6. POS Tagger- Design python program to perform part-of-speech tagging on the text scraped from a website. 7. Sentiment Analysis- Design a Python program to analyze the given text.	**6+4**	**(CO1) Apply**

Module-2: N-Grams		
N-Grams: Simple N-grams, Smoothing- Laplace smoothing, Good Turing Discounting, Backoff, Entropy. Morphology: Inflectional morphology, Derivational morphology. **Self-Study Component:** Advance issues in Good Turing discounting estimation. **Practical:** 1. N grams- Implement a Python program to implement N-Gram 2. Smoothing-Design a Python program to perform smoothing using various methods in Python. 3. Good turing- Develop a Python program to calculate good turing frequency.	**6+4**	**(CO2) Apply**
Module-3: Lexical Semantics		
Semantic: Meaning Representation, Lexical Semantics-Relationships, Ambiguity, Word Sense Disambiguation –Selectional Restriction-based word sense disambiguation, context-based word sense disambiguation Approaches-Lesk's Algorithm, Knowledge source in WSD. **Self-Study Component:** Context-based word sense disambiguation Approaches-KNN algorithm & Bayesian Classification. **Practical:** 1. Lexical Semantics- Design Python program to do text classification. 2. Meaning Representation- Implement a Python program to represent the meaning of the given text. 3. Disambiguity- Design the lesk algorithm in Python to handle word sensedisambiguation.	**6+4**	**(CO3) Apply**
Module-4: Information Retrieval		
Information Retrieval-Introduction, Design features of information retrieval systems- Indexing, eliminating stop words, Stemming, Classical information retrieval Models- Boolean model, Probabilistic model, Vector space model. **Applications:** Information Extraction, Automatic text summarization: Types of Summaries & Approaches, Question –Answer System: Architecture of an Open-Domain Question-Answering System. **Self-Study Component:** Topic Modelling. **Practical:** 1. Information Extraction- Design Python programs to extract structured information from unstructured information. 2. Filtering Stop Words- Implement a python program to filtering stop words. 3. Question Answering System- Design a questioning answer system using Python.	**6+4**	**(CO4) Apply**
Module-5: Language Processing using Large Language Model		
Introduction to GPT; Architecture - Attention mechanism, Queries, Keys and Values, Causal Masking, Transformer Block, Positional Encoding; Stochastic Text Generation with GPT: Training GPT, Analysis of Generated Text - Temperature parameter, Attention score for generated words. **Self-Study Component:** Overview of other large language models for different NLP tasks: BERT, T5, GPT-3, GPT-4, ChatGPT. **Practical:** 1. Casual Masking-Implement a python program to do casual masking in GPT. 2. Positional Encoding-Implement a python code to do positional encoding in GPT.	**6+4**	**(CO5) Apply**

Course outcomes:

The students will be able to

1. Apply the Natural Language processing concepts to the different applications of Corpus and perform word level analysis.
2. Solve the N grams problems using smoothing techniques.
3. Make use of meaning representation and word sense disambiguation models.
4. Build NLP models-Information retrieval, text summarization and question-answering system.
5. Apply NLP in large language model.

Reference Books:

1. Siddiqui T., Tiwary U. S. Natural language processing and Information retrieval, OUP, 2023.
2. Daniel Jurafsky and James H Martin, "Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", 2nd Edition, Prentice Hall, 2022.
3. David Foster. Transformers, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, Second Edition, O'Reilly, 2023.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Generative Artificial Intelligence (PCI)
Credit: 4

Course Code	23AML163	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

- get an introduction to generative AI and its landscape,
- learn generative adversarial networks as generative models,
- learn how large language models generate text,
- learn advanced techniques for generating text using large language models and
- learn techniques for generating images using diffusion models.

Prerequisites: Foundation of machine learning and deep learning

	Number of Hours	Bloom's Level
Module-1: Introduction		
Introduction to Generative Artificial Intelligence, Prompt Engineering and Principles of Prompting, Evaluating Quality of Generated Response; Introduction to Large Language Models (LLMs) for Text Generation: Vector Representation, Transformer Architecture, Probabilistic Text Generation; Different Pretrained Text Generation Models **Practical:** 1. Experiment in an appropriate prompt playground (e.g. OpenAI Playground) over different text prompts issued in different contexts with respect to principles of prompts and observe respective responses. 2. Repeat the same experiment for any appropriate image model to generate images in different prompt contexts. Self-Study: Quantization & LoRA, Model Comparison;	**6+4**	**(CO1) Apply**
Module-2: Generative Adversarial Networks		
Distinction between Autoencoder and Generative Adversarial Network (GAN); The Generator and Discriminator; Deep Convolutional GAN; Training and Evaluation of GAN; Wasserstein GAN with Gradient Penalty; Conditional GAN; Progressive Growing of GAN; **Practical:** 1. Implement a Conditional GAN model that learns to generate	**6+4**	**(CO2) Apply**

handwritten digits of choice. 2. Implement a StyleGAN to generate realistic looking images in an appropriate domain. Self-study: StyleGAN		
Module-3: Text Generation using Generative Pretrained LLMs		
Text Generation using OpenAI GPT: Generating List, Responses in Different Data Format, Language Translation, Context for Reasoning, Text Style Unbundling for Textual Features, Summarization, Chunking, Sentiment Analysis, Chain of Prompts (Least to Most), Role Prompting, Avoiding Hallucinations, Strategies for Classification with LLMs **Practical:** 1. Use an LLM to generate content in diverse formats (e.g. plaintext, JSON, YAML, mermaid, etc.) and extract information using appropriate methods. 2. Apply chain-of-prompts (least-to-most) technique for sequential generation and extraction of increasingly detailed knowledge from a LLM on a given topic. Self-Study: Criteria Evaluation, Meta Prompting	**6+4**	**(CO3) Apply**
Module-4: Advanced Techniques for Text Generation		
Text Generation with LangChain Framework: Chat Models, Prompt Templates, Expression Language, Output Parsers, Response Evaluation, Function Calling, Extracting Data, Query Planning, Few-shot Prompt Templates, Data Connection, Text Splitters, Task Decomposition, Prompt Chaining Text Generation with Embeddings & Vector Databases: Retrieval Augmented Generation (RAG) Patterns (Baseline RAG and GraphRAG), Embeddings, Document Loading, Storing and Querying Vector from a Vector Database (such as Pinecone, FAISS), RAG with LangChain, Self-querying; **Practical:** 1. Create an LLM application for an appropriate task using LangChain framework and RAG pattern. Self-Study: Fine-tuning LLMs with reinforcement learning from human feedback (RLHF)	**6+4**	**(CO4) Apply**
Module-5: Representation Learning using Autoencoders		

Introduction to Diffusion Models for Image Generation: OpenAI DALL-E, Midjourney, Stable Diffusion, Google Gemini; Standard practices for image generation - Format and art style modifiers, Reverse engineering prompts, Quality boosters, Negative prompts, Weighted terms, Prompting with an image **Safety in LLMs:** Introduction to risks associated with LLMs – Hallucinations, memorization, disinformation, harmful content, privacy; Risk mitigation with RLHF and RAG; **Practical:** 1. Experiment to generate images using Midjourney over all relevant standard prompting practices mentioned above and observing the relevant changes in model output. Self-Study: In and out painting, Prompt rewriting, Prompt analysis	**6+4**	**(CO5) Apply**

Course outcomes:

The students will be able to

1. experiment with a prompt playground to generate both text and images applying principles of prompts,
2. build and apply GAN models to generate images,
3. apply large language models for text generation against different NLP task requirements,
4. build an LLM application LangChain framework and RAG pattern, and
5. apply standard techniques to maximize the output and formats from diffusion models towards image generation.

Reference Books:

1. Numa Dhamani. *Introduction to Generative AI*, Manning, 2024 [1]
2. David Foster. *Generative Deep Learning*, Second Edition, O'Reilly, 2023 [2]
3. James Phoenix, Mike Taylor. *Prompt Engineering for Generative AI*, O'Reilly, 2024 [3, 4, 5]
4. Denis Rothman. *Transformers for Natural Language Processing and Computer Vision: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E3*, 3rd Edition, Packt Publications, 2024 [4, 5]
5. Ben Auffarth. *Generative AI with LangChain: Build Large Language Model (LLM) Apps with Python, ChatGPT, and Other LLMs*, Packt, 2023
6. Jakub Langr, Vladimir Bok. *GANs in Action*, Manning, 2019
7. Tanmoy Chakraborty. *Introduction to Large Language Models Generative AI for Text*, 2024, Wiley
8. Joseph Babcock and Raghav Bali. *Generative AI with Python and TensorFlow 2: Create Images, Text, and Music with VAEs, GANs, LSTMs, Transformer Models*, Packt, 2021
9. Zhihan Lyu. *Applications of Generative AI*, Springer, 2024

B. N. M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Image Processing and Computer Vision (PBL)
Credit: 2

Course Code	23AML164	**CIA Marks**		50
Teaching Hours/Week (L: T: P: J)	0:0:2:2	**SEA Marks**		50
Total Number of Contact Hours	50	**Exam Hours**		03

Course Learning Objectives:

This course will enable students to:

- Understand the fundamentals of image processing
- Understand the image transform, restoration techniques and methods used in digital image processing
- Understand the image enhancement techniques, Morphological Operations and Segmentation used in digital image processing
- Understand the various techniques used in computer vision
- Understand the image data compression and motion analysis of computer vision

Prerequisites: C Programming, Python Programming

Tool using: MATLAB R2024

Task 1	**Number of Hours**	**Bloom's Level**
Digital Image Fundamentals: Fundamentals of Digital Image Processing, Components of an Image Processing System, Image Sensing and Acquisition, Image Sampling and Quantization. **Practical using MATLAB** 1. Write a Program to read various formats of digital image and apply image sampling and quantization techniques. 2. Write a Program to read a digital image. Split and display image into 4 quadrants, up, down, right and left. **Self-Study component:** Understand the DIP concepts of real time imaging data and understand the various imaging file formats	**2**	**(CO1) Apply**
Task 2		
Spatial Domain: Some Basic Intensity Transformation Functions, Histogram Processing, **Practical using MATLAB** 1. Write a program to show rotation, scaling, and translation of an image. 2. Write a Program to Read an image and extract and display low-level features such as edges, textures using filtering techniques **Self-Study component:** Analyze the various imaging files using the functions of imread( ), imshow( ) and imtool( )	**2**	**(CO1) Apply**
Task 3		
Image Restoration and Reconstruction: A model of the Image Degradation/Restoration Process, Noise Models, **Practical using MATLAB** 1. Write a Program to Image Enhancement-Spatial filtering and frequency domain filtering 2. Write a Program to restore and reconstruct the image using special filtering **Self-Study component:** Study various noise Models of digital image processing	**2**	**(CO2) Apply**

Task 4		
Color Image Processing: Color Fundamentals, Color Models- RGB Color Model, **Practical using MATLAB** 1. Write a Program to analysis of images with different color models. 2. Write a Program to Image segmentation – Edge detection, line detection and point detection **Self-Study component:** Apply CMY and CMYK Color model and HSI Color Model.	**2**	**(CO2) Apply**
Task 5		
Morphological Image Processing: Preliminaries, Erosion and Dilation, **Practical using MATLAB** 1. Write a Program to Region based segmentation - clustering technique. 2. Write a Program on Image Compression algorithm **Self-Study component: Study** The Hit-or-Miss Transforms, Opening and Closing	**2**	**(CO3) Apply**
Task 6		
Image Segmentation: Fundamentals, Edge detection, Region Segmentation. **Practical using MATLAB** 1. Write a Program for human face detection using webcam. 2. Write a Program for detect the object. **Self-Study component:** Study on canny and sobel edge detection, Clustering and Super pixels.	**2**	**(CO3) Apply**
Task 7		
Introduction to Computer Vision: Image representation and Image analysis task, Cameras: An Overview- Photo sensitive sensors. **Practical using MATLAB** 1. Write a program to analysis of images with different color models. 2. Write a Program to extract the human facial features Eye, Nose, Mouth of an image. **Self-Study component:** Study the internal functions of digital camera	**2**	**(CO4) Apply**
Task 8		
3D Vision Geometry: 3D Vision tasks: Marr's theory, Basics projective geometry. **Practical using MATLAB** 1. Write a program to detect the single and group face of still images 2. Write a program to detect the single and group face using webcam video input **Self-Study component:** Study on Points and Hyperplanes in projective space	**2**	**(CO4) Apply**
Task 9		
Motion Analysis: Differential motion analysis methods. **Practical using MATLAB** 1. Write a Program to detect the motion of an object in an input video. 2. Write a program to find the motion of a person in video **Self-Study component:** Detection of specific motion pattern, Video tracking: background modeling, Kernel based object tracking.	**2**	**(CO5) Apply**
Task 10		
Optical flow: Optical flow computation. **Practical using MATLAB** 1. Write a Program to apply Histogram Processing to the video 2. Apply all filtering techniques on human action recognition activities **Self-Study component:** Optical flow in motion analysis and Image compression models	**2**	**(CO4) Apply**

Projects: Develop real world application using MATLAB graphical user interface or Python 1. Automated Real-Time Detection of Potentially Suspicious Behavior in Public Transport Areas 2. Brain Tumor Segmentation Based on Local Independent Projection-Based Classification 3. Emotional stress recognition system using EEG and psychophysiological signals 4. Image Contrast Enhancement Using Color and Depth Histograms 5. Medical image classification using PSO Optimization techniques 6. Fast and Accurate Detection and Classification of Plant Diseases 7. A High-Performance Fingerprint Matching System for Large Databases Based on GPU 8. Fruit classification using computer vision and feedforward neural network 9. Real Time Finger Print Segmentation Based on Local Independent Projection-Based Classification 10. A therapy System for Articulation Disorder correction 11. Evaluation of Color Spatio-Temporal Interest Points for Human Action Recognition 12. Visual Voice Activity Detection as A Help for Speech Source Separation from Convolutive Mixtures 13. Identification of human facial expression signal classification using Spatio Temporal Algorithm 14. Employee Attendance Monitoring using SURF Algorithms 15. Lung Cancer Detection in Chest X - Ray Image 16. Brain Cancer Detection using Machine Learning 17. Smart Digital Monitoring of Student's Attendance system using PCA 18. Development of Smart Voting system using Face and Finger data fusion analysis 19. Detect and recognize the Apple disease detection using machine learning models 20. Detection melanoma skin cancer detection using machine learning models	**(CO5) Create**

Course Outcomes: At the end of the course students should be able to:

1. Understand, Ascertain and describe the basics of image processing concepts through mathematical interpretation.
2. Apply image processing techniques in both the spatial and frequency (Fourier) domains.
3. Demonstrate image enhancement techniques, Morphological Operations and Segmentation used in digital image processing.
4. Conduct independent study and analysis of Image Enhancement techniques.
5. Apply computer vision techniques in image data compression and motion analysis of computer vision

Reference Books:

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Third Ed., Prentice Hall, 2008.
2. Sonka, Hlavac, Boyle, Digital Image Processing and Computer Vision, India Edition,
3. Digital Image Processing- S.Jayaraman, S.Esakkirajan, T.Veerakumar, TataMcGraw Hill 2014.
4. Fundamentals of Digital Image Processing-A. K. Jain, Pearson 2004.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Recommender Systems (Professional Elective I)
Credit: 3

Course Code	23AML16511	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

- To introduce the concept of recommender systems
- To be aware of various issues related to Personalization and Recommendations.
- To be aware of design space, trade-offs and its application in various domain
- To review basic approaches to building recommendations.

Prerequisites:

Machine learning & Advanced Machine Learning concepts

	Numberof Hours	**Bloom's Level**
Module-1		
An Introduction to Recommender System: Introduction, Goals of Recommender Systems, Basic Models of Recommender Systems, Domain-Specific Challenges in Recommender Systems, Advanced Topics and Applications. **Self-study component:** Pick a domain (e.g., healthcare, e-commerce) and list 3 unique RS challenges it faces. **RB1: Ch1**	**8**	**Understand (CO1)**
Module-2		
Neighborhood-Based Collaborative Filtering: Introduction, Key Properties of Ratings Matrices, Predicting Ratings with Neighborhood-Based Methods, Clustering and Neighborhood-Based Methods, Dimensionality Reduction and Neighborhood Methods, A Regression Modeling View of Neighborhood Methods, Graph Models for Neighborhood-Based Methods **Self-study component:** Given a dataset, try at least two neighborhood-based methods (e.g., item-based + graph model), evaluate and	**8**	**Analyze (CO2)**

compare. **RB1: Ch2**		
Module-3		
Model-Based Collaborative Filtering: Introduction, Decision and Regression Trees, Rule-Based Collaborative Filtering, Naive Bayes Collaborative Filtering, Latent Factor Models **Self-study component:** Build and compare at least two model-based approaches (e.g., Naive Bayes vs. Matrix Factorization) on the same dataset. **RB1: Ch3**	**8**	**Analyze (CO2, CO3)**
Module-4		
Content-Based Recommender Systems: Introduction, Basic Components of Content-Based Systems, Preprocessing and Feature Extraction, Learning User Profiles and Filtering, Content-Based Versus Collaborative Recommendations, Using Content-Based Models for Collaborative Filtering **Self-study component:** Identify 3 platforms using content-based recommendations (e.g., Goodreads, Spotify) and describe what features they may use. How does Netflix or Spotify combine approaches in real-world systems? **RB 1: Ch4**	**8**	**Analyze (CO2)**
Module-5		
Knowledge-Based Recommender Systems Introduction, Constraint-Based Recommender Systems, Case-based Recommender Systems **Evaluating Recommender Systems** Introduction, Evaluation Paradigms, General Goals of Evaluation Design, Design Issues in Offline Recommender Evaluation, Accuracy Metrics in Offline Evaluation, Limitations of Evaluation Measures **Self-study component:** Design 3 logical constraints for a travel recommender system. Create a slide deck comparing evaluation strategies with real-world examples (e.g., Netflix, Spotify, Amazon). **RB 2: Ch. 5.1,5.2,5,3,7**	**8**	**Analyze (CO2, CO4)**

Course outcomes:

The students will be able to

1. Understand the basic concepts of recommender systems
2. Solve mathematical optimization problems pertaining to recommender systems
3. Analyze the different approaches towards recommendation
4. Carry out performance evaluation of recommender systems based on various metrics

Reference Books:

1. Charu C. Aggarwal, Recommender Systems: The Textbook, Springer, 2016.
2. Dietmar Jannach , Markus Zanker , Alexander Felfernig and Gerhard Friedrich , Recommender Systems: An Introduction, Cambridge University Press (2011), 1st ed.
3. Francesco Ricci , Lior Rokach , Bracha Shapira , Recommender Sytems Handbook, 3rd edition, Springer (2022)

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Cryptography (Professional Elective –I)
Credit: 3

Course Code	23AML16541	CIE Marks:	50
Teaching Hours/Week (L: T: P: J)	3:0:0:0	SEE Marks:	50
Total Number of Lecture Hours	40	SEE Duration:	3 Hours

Course Learning Objectives:
This course will enable students to

- Enable to learn the fundamental concepts of network security, cryptography, and utilize these techniques in computing systems.
- Discuss about various encryption techniques.
- Understand the concept of public key Cryptography.
- Introduce and demonstrate message authentication, hash function, wireless security and internet security protocols.

Prerequisites:

- Knowledge of computer networks, operating systems, programming, and basic mathematics is essential to understand how data flows, how systems can be secured.

Module 1: BACKGROUND	**Number of Hours**	**Bloom's Level**
Computer, Network Security and Cryptography concepts: Computer Security Concepts The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms, Fundamental Security Design Principles , Attack Surfaces and Attack Trees. Classical Ciphers: Introduction to cryptography, cryptanalysis, and cryptology, Overview of cryptography. A Model for Network Security, Standards. **Self Study:** Transposition Methods	**08**	**(CO1) Apply**
Module 2: SYMMETRIC CIPHERS		
SYMMETRIC CIPHERS Advanced Encryption Standard: Finite Field Arithmetic, AES Structure, AES Transformation Functions ,AES Key Expansion , An AES Example, AES Implementation, **Block Cipher Operation :**Multiple Encryption and Triple DES , Electronic Codebook ,Stream Ciphers. Cipher Block Chaining Format-Preserving Encryption **Self Study:** Advanced symmetric encryption techniques	**08**	**(CO2) Apply**
Module 3: ASYMMETRIC CIPHERS		

Public-Key Cryptography and RSA : Principles of Public-Key Cryptosystems : The RSA Algorithm **Other Public-Key Cryptosystems :**Diffie-Hellman Key Exchange ,Elgamal Cryptographic System ,Elliptic Curve, Elliptic Curve Cryptography, Pseudorandom Number Generation Based on an Asymmetric Cipher **Self Study:** Public-key cryptography fundamentals	**08**	**(CO3) Apply**
Module 4: CRYPTOGRAPHIC DATA INTEGRITY ALGORITHMS		
Cryptographic Hash Functions: Applications of Cryptographic Hash Functions ,Two Simple Hash Functions, Requirements and Security ,Hash Functions Based on Cipher Block Chaining ,Secure Hash Algorithm (SHA) ,SHA-3 .**Message Authentication Codes** :Message Authentication Requirements message Authentication Functions, Requirements for Message Authentication Codes , **Digital Signatures** : Digital Signatures. Elgamal Digital Signature Scheme , Elliptic Curve Digital Signature Algorithm. **Self Study:** Digital signature algorithms including SHA variants.	**08**	**(CO4) Apply**
Module 5: NETWORK AND INTERNET SECURITY		
Cloud Computing, Cloud Security Risks and Countermeasures, Data Protection in the Cloud, Cloud Security as a Service Addressing Cloud Computing Security Concerns **Wireless Network Security:** Wireless Security, Mobile Device Security, **Electronic Mail Security:** Email Threats and Comprehensive Email Security, S/MIME **IP Security** :IP Security Overview, IP Security Policy, Internet Key Exchange **Self Study:** IP security mechanisms.	**08**	**(CO5) Apply**

Course Outcomes:

At the end of the course, the students will be able to:

1. Identify the cryptographic solutions for the problems.
2. Classify and apply the symmetric encryption techniques to solve the problems.
3. Apply and illustrate various public key cryptographic techniques to solve the problems.
4. Evaluate the authentication and hash algorithms.
5. Apply wireless security and internet security protocols.

Reference Books

1. Cryptography and Network Security: Principles and Practice, 8th Edition, Pearson Education by William Stallings
2. Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Education by William Stallings
3. Cryptography and Network Security, Behrouz A. Foruzan, 3rd edition, Tata McGraw Hill, 2007
4. Cyber Defense Mechanisms: Security, Privacy, and Challenges (Artificial Intelligence (AI): Elementary to Advanced Practices) by Gautam Kumar (Editor), Dinesh Kumar Saini, Nguyen Ha Huy Cuong, 1st edition, CRC Press, 2020.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Data Warehousing & Data Mining (Professional Elective I)
Credit: 3

Course Code	23AML16551	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
1. Define multi-dimensional data models.
2. Explain rules related to association, classification and clustering analysis.
3. Compare and contrast between different classification and clustering algorithms

Prerequisite: DBMS, working with datasets

	Number of Hours	Bloom's Level
Module-1 Data Warehouse: Basic Concepts and Modeling		
Basic Concepts: Introduction, A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Metadata Repository. **Data Cube:** A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations, Starnet Query Model for Querying multidimensional databases. Self-Study: Explore Data warehousing tools	**8**	**(CO1) Apply**
Module-2 Data Warehouse: Design, Usage & Implementation		
Design, Usage: A Business Analysis Framework for data warehouse design, Data Warehouse design process, data warehouse usage for Information processing, Online analytics processing to Multidimensional Data Mining. **Implementation:** Efficient Data cube Computation: An Overview, Indexing OLAP Data: Bitmap Index and Join Index, Efficient Processing of OLAP Queries, OLAP server architectures: ROLAP versus MOLAP versus HOLAP. Comprehensive Concepts of a **Data Lake** Self-Study: Working of a Data Lake Lambda Architecture	**8**	**(CO1) Apply**
Module-3 Data Mining: Introduction & Association Analysis		
Introduction: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity.	**8**	**(CO2) Apply**

Association Analysis: Problem definition, Frequent Itemset Generation, Rule Generation, Compact Representation of Frequent Itemsets, Alternative methods for generating frequent itemsets, FP-Growth algorithm Self-Study: Sequential Patterns		
Module-4 Clustering		
Prototype- Based Clustering: Fuzzy Clustering, Clustering using mixture models, Self-Organizing Maps (SOM) **Density-based clustering:** Grid-based, subspace, DENCLUE Self-Study: Agglomerative Hierarchical clustering	**8**	**(CO3) Apply**
Module-5 Clustering-II		
Graph- Based Clustering: Sparsification, Minimum Spanning tree clustering, OPOSSUM: Optimal partitioning of Sparse similarities using METIS, Chameleon: Hierarchical clustering with dynamic modeling, shared nearest neighbor similarity, The Jarvis-Patrick clustering algorithm, SNN Density, SNN-Density Based clustering, **Scalable Clustering algorithms:** General Issues and approaches, BIRCH, CURE. Self-Study: Anomaly Detection based on proximity	**8**	**(CO3) Apply**
Course outcomes: 1. Identify data mining problems and implement the data warehouse 2. Write association rules for a given data pattern. 3. Choose between classification and clustering solutions.		
Reference Books: 1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression, 2014. 2. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012. 3. Sam Anahory, Dennis Murray: Data Warehousing in the Real World, Pearson, Tenth Impression, 2012. 4. Michael. J. Berry, Gordon.S. Linoff: Mastering Data Mining , Wiley Edition, second edtion, 2012. 5. Tomcy John, Pankaj Misra : Data Lake for Enterprise, 2017 Packt Publishing.		

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER –VI

Responsible AI (Professional Elective-II)
Credit:3

Course Code	**23AML16612**	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

1. Responsible AI and Ethics
2. Ethics and Social implications of AI.
3. Principles of Responsible AI and machine learning
4. Global AI governance framework and standards of OECD
5. Legal and regulatory landscape for AI, covering laws related to non-discrimination, data protection and intellectual property

Prerequisites: Basics of AI and ML

	Number of Hours	**Bloom's Level**
Module-1: Introduction to Responsible AI and its Principles, Fundamentals of AI		
Responsible AI, Principles, AI Framework, Bias in AI, RAI Policies and Governance, Creating and Evaluating AI models, Identify Ethical issues, Issue spotting and mitigation, monitoring and optimizing AI model Types of AI Systems: Narrow vs. General AI, Machine Learning Basics and Training Methods, Deep Learning, Generative AI, and Transformer Models, Natural Language Processing and Multi-modal Models, Socio-technical AI Systems and Cross-disciplinary Collaboration, The History and Evolution of AI and Data Science	**8**	**(CO1) Apply**
Module-2 Understand AI impact on society, RAI principles trustworthy AI		
Individual Harms: Civil Rights, Safety, and Economic Impact, Group Harms: Discrimination and Bias in AI Systems, Societal Harms: Democracy, Education, and Public Trust, Organizational Risks: Reputational, Cultural, and Economic Threats, Environmental and Ecosystem Impacts of AI, Redistribution of Jobs and Economic Opportunities Due to AI, AI's Impact on Workforce and Educational Access Core Principles of Responsible AI, Human-centric AI Systems, Transparency, Explainability, and Accountability in AI, Safe, Secure, and Resilient AI Systems,	**8**	**(CO2) Apply**

Privacy-Enhanced AI Systems and Data Protection, OECD and EU Standards for Trustworthy AI, Comparison of Global Ethical Guidelines for AI		
Module-3: AI laws and regulatory compliance, AI Legal Framework, GDRP Requirements		
Section Introduction, AI-Specific Laws and Regulations, Non-Discrimination Laws and AI Applications, Product Safety Laws for AI Systems, Privacy and Data Protection in AI Systems, Intellectual Property and AI: Legal Considerations, Key Components of the EU Digital Services Act , the Intersection of AI and GDPR Requirements Comparison of Global Ethical Guidelines for AI, Requirements for High-Risk AI Systems and Foundation Models, Notification and Enforcement Mechanisms under the EU AI Act, Canada's Artificial Intelligence and Data Act (Bill C-27), Key Components of U.S. AI-related State Laws, China's Draft Regulations on Generative AI, Harmonizing Global AI Laws and Risk Management Frameworks	**8**	**(CO3) Apply**
Module-4: AI Development Life Cycle and Testing		
Section introduction, Determining AI governance structure and responsibilities, Data strategies, model selection, Ethical design in AI system and Architecture, Understanding the Governance Challenges in AI Planning, Cross-functional Team Collaboration in AI Planning Feature engineering in AI model, Model Training and best practices, Testing and Validation process AI Models, Testing AI Models with Edge Cases and Adversarial Inputs, Privacy-preserving Machine Learning Techniques, Repeatability Assessments and Model Fact Sheets, conducting Algorithm Impact Assessments	**8**	**(CO4) Apply**
Module-5: Implementation of AI Governance, AI Project Management and Risk Management		
Section Introduction, Creating AI Risk Management Frameworks, AI Governance Infrastructure: Key Roles and Responsibilities, Cross-functional Collaboration in AI Governance, AI Regulatory Requirements and Compliance Procedures, Establishing a Responsible AI Culture within Organizations, Assessing AI Maturity Levels in Business Functions, Managing Third-Party Risks in AI Systems Scoping AI Projects: Identifying Key Objectives, Mapping AI Risks: Identifying Internal and External Threats, Developing Risk Mitigation Strategies for AI Projects, Constructing a Harms Matrix for AI Risk Assessment, Conducting Algorithm Impact Assessments, Engaging Stakeholders in AI Risk Management, Data Provenance, Lineage, and Accuracy in AI Systems	**8**	**(CO5) Apply**

Course Outcomes:

The students will be able to

1. Understand Responsible AI ,its Principles and AI Life cycle
2. Apply fundamentals of AI and its impact into society
3. Apply AI laws, regulatory compliance, AI Legal Framework,
4. Apply AI Governance and Risk Management
5. Apply OECD standards in the AI product life cycle

Reference Link:

https://lms.simplilearn.com/courses/6752/Introduction-to-Responsible-AI/syllabus
https://www.simplilearn.com/free-google-cloud-responsible-ai-course-skillup?tag=ethical%20AI
https://www.udemy.com/course/ai-governance-professional-aigp-certification-ai-mastery/?couponCode=JUST4U02223

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER –VI

Cyber Security and Privacy (Professional Elective-II)
Credit:3

Course Code	23AML16642	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

- Understand fundamentals of Cyber security.
- Explain security Architecture and security techniques
- Illustrate risk management issues and solutions.
- Explain data privacy and protection.

Prerequisites: Cryptography fundamentals

	Number of Hours	Bloom's Level
Module-1: Introduction to Cyber Security		
Introduction: Fundamentals of Security, Threat actors, physical Security, Social Engineering, Malware.	**8**	**(CO1) Apply**
Module-2: Risk Management		
Cryptographic solutions, Risk management, Third party vendor risk management, Governance and Compliance. Asset and change management, Cyber Resilience and Redundancy	**8**	**(CO2) Apply**
Module-3: Architecture		
Security Architecture, Security Infrastructure, identity and access management (IAM) Solution, Vulnerabilities and attacks, Malicious Activity.	**8**	**(CO3) Apply**
Module-4: Security Techniques		
Hardening, Security Techniques, Vulnerability Management, Alerting and Monitoring, Incident Response, Investing an Incident, Automation and orchestration, Security Awareness.	**8**	**(CO4) Apply**
Module-5: Data Privacy and Protection		
Data privacy - Data Protection, Complete GDPR, GDPR Certification, privacy, Hands on Programs.	**8**	**(CO5)**

		Apply

Course Outcomes:

The students will be able to

1. Understand the Fundamental Cyber Security Concept.
2. Apply security risk management strategies.
3. Identify different types of attack, vulnerability issues and Malicious Activity.
4. Apply different security techniques to address security issues.
5. Make use of GDPR for data Privacy and Protection.

Reference Books:

1. Udemy course: https://www.udemy.com/share/101Wj83@I57HMbLebSbmoRf1WCIXshoCztsg0Vo2PMWANE92oMcvsCK9YVpf-YTnK604c1JQ/
2. Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Education by William Stallings
3. Cryptography and Network Security, Behrouz A.Foruzan, 3rd edition, Tata McGraw Hill, 2007
4. Cyber Defense Mechanisms: Security, Privacy, and Challenges (Artificial Intelligence (AI): Elementary to Advanced Practices) by Gautam Kumar (Editor), Dinesh Kumar Saini, Nguyen Ha Huy Cuong, 1st edition, CRC Press, 2020

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VI

Data Science & Data Analytics (Professional Elective II)
Credit: 3

Course Code	23AML16652	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	3:0:0:0 AML	SEA Marks	50
Total Number of Lecture Hours	40	Exam Hours	03

Course Learning Objectives:
This course will enable students to

- Introduce the concepts of python programming, SQL and HTML.
- Identify different cloud platforms and their characteristics for data specific core computations.
- Identify the concepts of data warehousing and big data.
- Determine the concepts of Power Business Intelligence.
- Design queries for analysis of data.

	Number of Hours	**Bloom's Level**
Module-1		
Introduction to Data Science- Introduction, Definition, Data Science in various fields, Impact of Data Science, Data Analytics Life Cycle, Data Science Toolkit, Data Scientist, Introduction, Types of Data, Classification of digital Data, Sources of Data **SQL**: Database and RDBMS, NoSQL, RDBMS Advanced Queries, Functions, Aggregate and Analytical, Data Integrity, Subqueries & Views, SQL Advanced Concepts **Python:** Introduction to python, python variables and syntax, conditions, python strings, tuples, functions in python, Matplotlib and seaborn for data visualization, Data visualization: plots, histograms, heatmap, Data preprocessing: missing values, outliers, encoding, Scaling and normalization, multivariate analysis, packages and python connectivity **XML, Javascript and Webservices:** What is XML, XML basic tags and XML examples, What is Javascript and why Javascript Requires and Javascript example, Introduction to Webservices, What is Client and server technologies, HTTP Protocols and various methods of HTPP?	**8**	**Apply**
Module-2		
Data specific core: AWS Data Services: Identity Access Management- IAM, Elastic Cloud Compute- EC2, AWS RDS, AWS Glue, AWS Redshift, AWS EMR, AWS - Dynamo DB, AWS QuickSight, AWS SageMaker **GCP Data Services**: Compute Engine, Cloud Storage, Cloud SQL, Spanner, Datastore, Bigtable, BigQuery, Data Proc, DataFlow, Cloud Composer, Dataprep, Data Fusion, Cloud AutoML, Looker	**8**	**Apply**

Azure Data Services: introduction to Azure storage, benefits of Azure storage, **Blob storage, Azure files, Azure container storage**, types of storage accounts, **encryption**, **Azure HDInsight**, What is HDInsight and the Hadoop technology stack?, Cluster types in HDInsight, Programming languages in HDInsight, Azure Cosmos DB - Database for the AI Era, Simplified application development		
Module-3		
Data Specific - DWH, DI/BI Concepts: Introduction to NoSQL, Key Features of NoSQL, advantages and disadvantages of NoSQL, Types of NoSQL database, Databricks architecture overview, High-level architecture, Serverless compute plane, Classic compute plane **Big Data Hadoop and Spark:** History and timeline of big data, RDDs vs DataFrames and Datasets, When to use them and why, Benefits of Dataset APIs, Understanding Hadoop Architecture, Components, and How It Works, Functional Programming.	**8**	**Apply**
Module-4		
Power BI: What is PowerBI, the parts of Power BI, How Power BI matches the role in a team or project, The flow of work in Power BI, How Microsoft Fabric works with Power BI, Paginated reports in the Power BI service, On-premises reporting with Power BI Report Server, powerBI visuals, Preattentive Attributes in Visualization	**8**	**Apply**
Module-5		
Snowflake: Getting started with snowflake, key concepts and architecture, supporting cloud platforms, supported cloud regions, snowflake editions, snowflake releases, overview of key features, overview of data life cycle, continuous data protection, snowflake ecosystem, snowflake partner connect, general configuration, snowflake architecture, snowflake virtual warehouse overview, snowflake features, Analysis of real time applications through snowflake.	**8**	**Analyze**
Course outcomes: The students will be able to: • Apply the core concepts of SQL, Python and XML to perform analytics on data. (Apply) • Develop a comparison study of various cloud platforms and study their services and database. (Apply) • Apply the concept of Data warehousing, business intelligence and big data to analyze datasets. (Apply) • Design and develop power business intelligence and visualization for data analytics through snowflake platform. (Apply) • Analyze datasets with appropriate programming language, SQL and snowflake queries. (Analyze)		

References:

Sl No	Modules	Recommended Links for reference
1	Data Specific - SQL Deep-dive	https://www.slideshare.net/search?searchfrom=header&q=SQL
2		https://docs.snowflake.com/en/sql-reference-commands
3		https://comparecloud.in/

4	Data Specific - DWH, DI/BI Concepts	https://www.geeksforgeeks.org/introduction-to-nosql/
5		https://docs.databricks.com/en/getting-started/overview.html
6		https://docs.snowflake.com/en/sql-reference-commands
7	Data Specific - Big Data Hadoop + Spark	https://www.techtarget.com/whatis/feature/A-history-and-timeline-of-big-data
8		https://www.databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html
9		https://medium.com/@chenglong.w1/demystifying-yarn-understanding-its-architecture-components-and-how-it-works-738dd95ad453
10		https://github.com/readme/guides/functional-programming-basics
11	AWS Data Services	https://aws.amazon.com/quickstart/
12		https://aws.amazon.com/about-aws/global-infrastructure/?p=ngi&loc=0
13	Azure Data Services	https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction
14		https://learn.microsoft.com/en-us/training/paths/azure-sql-fundamentals/
15		https://learn.microsoft.com/en-us/azure/data-factory/quickstart-get-started
16		https://learn.microsoft.com/en-us/azure/hdinsight/hdinsight-overview
17		https://learn.microsoft.com/en-us/azure/synapse-analytics/overview-what-is
18		https://learn.microsoft.com/en-us/fabric/get-started/microsoft-fabric-overview
19		https://learn.microsoft.com/en-us/azure/cosmos-db/introduction
20		https://learn.microsoft.com/en-IN/azure/machine-learning/tutorial-azure-ml-in-a-day?view=azureml-api-2
21	GCP Data Services	https://www.youtube.com/@googlecloudtech
22		https://thecloudgirl.dev/sketchnote.html
23		https://cloud.google.com/docs
24		https://cloud.google.com/architecture
25	Power BI	https://www.youtube.com/watch?v=yKTSLffVGbk
26		https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview
27		https://powerbi.microsoft.com/en-us/blog/
28		https://powerbi.microsoft.com/en-my/search/community/
29		https://www.youtube.com/watch?v=77jIzgvCIYY
30		https://appsource.microsoft.com/en-us/marketplace/apps?product=power-bi-visuals
31		https://www.perceptualedge.com/about.php
32		https://daydreamingnumbers.com/blog/preattentive-attributes-example/
33		https://www.storytellingwithdata.com/
34	Snowflake	https://docs.snowflake.com/en/sql-reference-commands
35		https://learn.snowflake.com/en/
36		https://quickstarts.snowflake.com/
37		https://www.snowflake.com/resource/7-snowflake-reference-architectures-application-builders/#main-content
38		https://www.snowflake.com/en/data-cloud/pricing-options/

B. N. M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

Agentic Artificial Intelligence (PCI)

SEMESTER– VII

Course Code	**23AML171**	**CIE Marks:**	**50**
Teaching Hours/Week (L: T: P: J)	**2:0:2:0**	**SEE Marks:**	**50**
Total Number of Lecture Hours	**50**	**SEE Duration:**	**3 Hours**

Course Learning Objectives:

This course will enable students to

- Experiment with the transition from static LLM prompting to autonomous Agentic AI, understand its core architecture and design aspects.
- Apply key agentic design patterns, tools use and context engineering to enable agents to solve complex, multi-step tasks autonomously.
- Analyse specialized frameworks like LangGraph to develop stateful and memory-aware agentic applications focusing on state management and short/long-term memory schemas
- Design multi-agent scenario, understand architecture and orchestration patterns, and create workflows
- Evaluate agentic workflows by implementing observability and monitor agents' performance.

Prerequisite: AI, Generative AI, Python programming

Module 1: INTRODUCTION TO AGENTIC AI	**Number of Hours**	**Bloom's Level**
Introduction to Agentic AI: Defining AI Agents, Agent Characteristics, Agents vs. Assistants vs. Models, Types of Agents, Access to Generative Models and Selection, Use Cases and Applications, Agentic Workflows vs. Traditional Workflows, Agentic AI Frameworks; **Designing Agentic AI Systems:** Core Components (Models, Tools, Memory, Orchestration), Design Principles, Design Trade-offs, Architecture Design Patterns, User Experience Design - Interaction Modalities, Communication Models (synchronous and asynchronous), Context Retention and Continuity, Communicating Agent Capabilities; **Practical:** 1. Create a minimal agent (a 'Hello' agent) that should be able to respond to generic questions and run it in both non-streaming and streaming mode. 2. Create a simple customer support agent that should be able to change the delivery address, cancel, refund for an order and confirm back to the user with all the updated details.	**10**	**Apply**
Module 2: TOOLS USE		
Tool types – Local Tools, APIs, Plug-ins, Model Context Protocol (MCP), Automated Tool Development; Orchestration – Agent Types (Reflex, ReAct, Planner-Executor, Query-Decomposition, Reflection and Deep Research), Tools Selection (Standard, Semantic and Hierarchical), Tools Execution, Topologies (Single, Parallel, Chains and Graph Execution), Agents as Tools, Context Engineering; Human in the Loop (HITL);	**10**	**Apply**

Practical: 3. Create an agent with a set of standard tools with a foundation model for it to select appropriate tools for a given context. 4. Implement an agent with API-based tools to enquire about additional information by performing web searches. 5. Implement a simple MCP server that evaluates arithmetic expressions and provides weather information, and an agent to call appropriate tools exposed by the MCP server based on the user's message.		
Module 3: KNOWLEDGE, MEMORY & LEARNING		
Knowledge & Memory: Foundational Approaches to Memory – Context Windows, Full-text Search; Short-term Memory with Agent Context and Long-term Semantic Memory with RAG (vector stores); GraphRAG – Using Knowledge Graphs and Dynamic Knowledge Graph, Note-taking **Learning in Agentic Systems:** Non-parametric Learning – Exemplar learning, Reflexion and Experiential learning; Parametric Learning – Fine-tuning Models, Supervised Fine-tuning, Direct Preference Optimization, Reinforcement Learning with Verifiable Rewards; **Practical:** 6. Implement memory by integrating a standard in-memory context provider into an agent. 7. Implement an agent that queries a knowledge graph to retrieve structured information against a user's question, combines it with a foundation model and responds with contextually rich informed answers.	**10**	**Analyze**
Module 4: MULTI-AGENT SYSTEMS		
Introduction to Multi-agent System: Single- vs. Multiagent Scenarios, Principles of Multi-agent Systems; Coordinations Patterns (Democratic, Manager, Hierarchical and Actor-Critic); Communication Technique – Local and Distributed (Agent-to-Agent protocol); Message Brokers and Event Buses; Automated Design of Agent Systems (ADAS); **Orchestration & Workflows:** Orchestration Patterns – Workflow as Deterministic Computation Graph (Sequential, Conditional and Parallel Execution) and Autonomous (Plan-based, Round-robin, Model-driven); Task Management Patterns; Workflows with Serialization and Checkpointing; **Practical:** 8. Build a system of multiple specialized agents, each equipped with appropriate tools, that collaborate to provide a solution for a relatively complex task around the supply-chain logistics domain. 9. Build a multiagent multi-step workflow orchestration demonstrating task sequencing, retry handling and failure management ensuring durability and reliability of the workflows for a suitable use case.	**10**	**Create**

Module 5: EVALUATION & MONITORING		
Evaluation: Key Considerations in Evaluation-Driven Development, Enabling Observability for Evaluation, Answer Extraction from Trajectory, Multimodal Evaluation, Deterministic Metrics and LLM-judges, Reference-free and Reference-based Metrics, Scoring Strategies; **Monitoring:** Monitoring Stacks and Selection; Instrumentation and Visualization, Monitoring Agent Performance (over error rate, latency, tool calls, cost, task completions and ethical metrics) and Scalability; **Practical:** 10. Implement evaluation for an appropriate autonomous multi-agent AI system against all relevant established evaluation metrics. 11. Extend the above experiment to implement an appropriate optimization loop (measure-analyze-modify-validate) by adjusting parameters such as instructions, tools, and orchestration patterns. 12. Implement observability with visualization in the above system using a right monitoring stack.	**10**	**Evaluate**

Course Outcomes:

At the end of the course, the students will be able to:

1. Build agents that connect to language models and support multi-turn conversations with user.
2. Construct agents that follow different orchestration patterns and use appropriate types of tools including MCPs.
3. Analyze the role of short-term and long-term memory and knowledge graphs in building context-aware AI agents.
4. Create multi-agent systems by implementing orchestration as deterministic graphs or autonomous and establishing communication amongst themselves over standard local and distributed protocols.
5. Evaluate the performance and reliability of agentic AI workflows using observability, metrics, and visualization techniques.

Reference Books

1. Michael Albada. *Building Applications with AI Agents: Designing and Implementing Multiagent Systems*, O'Reilly, 2025
2. Victor Dibia. *Designing Multi-Agent Systems: Principles, Patterns, and Implementation for AI Agents*, 2025
3. Micheal Lanham. *AI Agents in Action: Intelligent workflows with LLMs, MCP, A2A, and more*, Second Edition, Manning, 2026.
   "Micheal Lanham. *AI Agents in Action: Build, orchestrate, and deploy autonomous multi-agent systems*, Manning, 2025"
4. Valliappa Lakshmanan, Hannes Hapke. *Generative AI Design Patterns: Solutions to Common Challenges When Building GenAI Agents and Applications*, O'Reilly, 2025
5. The 2025 Guide to AI Agents: Your one-stop resource for gaining in-depth knowledge and hands-on applications of AI agents". IBM, https://www.ibm.com/think/ai-agents (accessed Jul. 08, 2025).
6. Agent Framework documentation. Microsoft, https://learn.microsoft.com/en-us/agent-framework/ (accessed May. 07, 2026)

B.N.M. Institute of Technology

An Autonomous Institution under VTU

Department of Artificial Intelligence and Machine Learning

Blockchain Technology (Professional Elective –III)

SEMESTER– VII

Course Code	23AML17243	CIE Marks:	50
Teaching Hours/Week (L: T: P: J)	3:0:0:0	SEE Marks:	50
Total Number of Lecture Hours	40	SEE Duration:	3 Hours

Course Learning Objectives:

This course will enable students to

- Conceptual understanding of the function of Blockchain as a method of securing distributed ledgers, how consensus on their contents is achieved, and the new applications that they enable.
- Apply Blockchain operations as distributed data structures and decision-making systems, their functionality and different architecture types.
- Evaluate existing "smart contract" capabilities and platforms, and examine their future directions, opportunities, risks and challenges.

Prerequisites: cryptography, data structures & programming, networks & distributed computing

Module 1	**Number of Hours**	**Bloom's Level**
INTRODUCTION The Chain and the Longest Chain, Cryptocurrency to Blockchain, Permissioned Model of Blockchain, what is Blockchain, Public Ledgers, Blockchain as Public Ledgers, Block in a Blockchain, Transactions, Distributed Consensus, Cryptographic Hash Function, Properties of a hash function, Hash pointer and Merkle tree, Digital Signature, Public Key Cryptography. **Self-Study Components: Type of nodes, Tokens, Gas, Various cryptocurrencies**	**8**	**Apply (CO1)**
Module 2		
Bitcoin and its Consensus-1 A basic cryptocurrency, Creation of coins, Payments and double spending, Bitcoin P2P Network, Transaction in Bitcoin Network, Block Mining, Block propagation and block relay, Why Consensus, Distributed consensus in open environments, Consensus in a Bitcoin network. **Self-Study Components: Ledger Design, SHA 128, SHA-256, SHA-512**	**8**	**Apply (CO2)**
Module 3		
Consensus-2 Bitcoin Consensus, Proof of Work (PoW) – basic introduction, Hashcash PoW, Bitcoin PoW, Attacks on PoW and the monopoly problem, Proof of Stake, Proof of Burn and Proof of Elapsed Time, The life of a Bitcoin Miner, Mining Algorithm, Mining Difficulty, Mining Pool, Permissioned model and use cases.	**8**	**Apply (CO3)**

Self-Study Components: Delegated Proof of stake, Proof of Authority, RAFT, PAXOS		
Module 4		
Smart contracts: Smart contract introduction and implementation, Solidity Programming basics, Financial Services, Crowdfunding, Smart Property, Blockchain 2.0 Protocol Projects Wallet Development Projects, Blockchain Development Platforms and APIs. DApps, DOAs. **Self-Study Components: Application design with smart contract in Remix IDE, Blockchain-as-a-Service (BaaS)**	**8**	**Analyze (CO4)**
Module 5		
Development Platforms Ethereum, Hyperledger Fabric: Transaction Flow, Hyperledger Fabric Details, Fabric – Membership and Identity Management, Hyperledger Fabric Network Setup, Blockchain in Financial Service, Blockchain in Supply, Blockchain in Government. **Self-Study Components: Blockchain and smart contract vulnerabilities**	**8**	**Apply (CO5)**

Course Outcomes:

At the end of the course, the students will be able to:

1. Apply the concepts and structure of Blockchain to identify suitable deployment scenarios over a traditional distributed database.
2. Apply Cryptographic functions to ensure the Blockchain properties to solve real time problems.
3. Apply the Consensus mechanisms to find better solutions for the problems.
4. Design and Analyze smart contracts, its potential usage and its limitations.
5. Design the Blockchain using any development platform along with test networks and supporting environment.

Reference Books

1. Blockchain by Melanie Swa, First edition, O'Reilly, 2015
2. Mastering Bitcoin: Unlocking Digital Cryptocurrencies by Andreas Antonopoulos, First edition, O'Reilly, 2014.
3. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits, Redbook by IBM, 2017.
4. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric.
5. Practical Artificial Intelligence and Blockchain by Ganesh Prasad Kumble, Anantha Krishnan, 2020, Packt Publishing, ISBN: 9781838822293.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VII

Predictive Analytics & Business Intelligence (Professional Elective III)
Credit: 3

Course Code	23AML17253	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	3hours

Course Learning Objectives:

This course will enable students to

- Understand and apply techniques to forecast future outcomes and make data-driven decisions
- Improve strategic decision making in various business areas like marketing, sales, and operations.

Prerequisites: Basics of ML model workflow

	Number of Hours	**Bloom's Level**
Module-1		
Decision Support: Introduction, Changing Business Environments, Computerized Decision Support, Managerial Decision Making, An Early Framework for Computerized Decision Support, Concepts of DSS. **Foundations and technologies for Decision Making:** Introduction and definitions of decision making, Phases of Decision making, how decisions are supported, Components of DSS. Self-study: Application Cases.	8	**CO1 (Apply)**
Module-2		
Techniques for predictive modeling: Data Warehousing Definitions and Concepts, Data Integration, Extraction, Transformation, and Load (ETL). Data Mining concepts, data mining process, data mining methods and software tools. Application of neural network-based system, support vector machines and nearest neighbor method for prediction. Self-study: Text mining applications and tools.	**8**	**CO2 (**Analyze**)**
Module-3		
Web Analytics, Web Mining and Social Analytics: Web Mining Overview, Web content and web structure mining, search engines, Search engine optimization, web analytics metrics and tools. Social Network Analysis and its metrics, Introduction to social media, social media analytics. Self study: Application Cases.	**8**	**CO3** **(**Analyze**)**
Module-4		
Business Analytics: A framework for Business Intelligence, Business Analytics overview. Business Reporting, Visual Analytics: Business Reporting Definitions and Concepts, Data and Information visualization, Different types of graphs and charts, the Emergence of Data Visualization and Visual Analytics, Performance Dashboards using Tableau.	**8**	**CO4 (**Analyze**)**

Case study: Saudi telecom company excels with Information visualization. Real time applications using BI tools: Tableau, Google looker studio. Self-study: Real time applications and use cases		

Module-5		
Emerging trends and Future Impacts of Business Analytics: Location-Based Analytics for Organizations, Analytics Applications for Consumers, Recommendation engines, Impacts of Analytics in Organizations, Issues in legality, privacy and ethics, An Overview of the Analytics Ecosystem. Self-study: cloud computing and BI	**8**	**CO5 (Apply)**

Course outcomes:

The students will be able to

1. Apply the basics of data and business to understand Decision Support systems
2. Analyze Extraction, Transformation, and Load (ETL) processes and predictive modeling techniques for efficient data preparation and predictive analytics in business intelligence.
3. Analyze web analytics and social media data to generate insights for business decision-making.
4. Analyze data using Business Intelligence (BI) tools and create interactive visualizations to support business decision-making.
5. Apply emerging trends for strategic decision-making to improve organizational performance.

Reference Books:

1. "Business Intelligence and Analytics: Systems for decision support", Ramesh Sharda, Dursun Delden, Efraim Turban, Pearson, 10th edition.
2. Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software by Dan Murray

Online Resources:

1. https://tinyurl.com/ycyc3v4t
2. https://www.tableau.com/products/tableau

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER –VII

Ethical Hacking (PE-IV MOOC)
Credit:3

Course Code	**23AML17344**	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

1. Mastery of Ethical Hacking Fundamentals
2. Proficiency in Information Gathering and Scanning Techniques
3. Expertise in Penetration Testing and Wireless Security
4. Advanced Exploitation and Scripting Skills
5. Development and Application of Malware and Security Tools.

Prerequisites: Basic computer and networking knowledge, programming and scripting skills, familiarity with operating systems and security concepts, foundational math and logic and awareness of cyber security threats, and ethical considerations.

	Number of Hours	**Bloom's Level**
Module 1: Introduction to Ethical Hacking and Environment Setup		
Introduction and Installation: Kali Linux in a Virtual Machine Setup, Virtual Box Installation, Kali Linux Installation, Enabling Full Screen. **Basic Linux Commands.** **Creating a Secure Hacking Environment:** Changing IP address and wireless adapter setup, creating a bootable Kali USB drive, Essential networking terms, Essential hacking terms, Additional Kali tools to install, Changing our MAC address with macchanger. **Footprinting:** Google hacking, Nikto basics, Whois tool, Email harvesting, Shodan, Zone transfer with Dig **Scanning:** Nmap part 1, 2 and 3, Zenmap, TCP scans, Nmap bypassing defences Nmap scripts part 1 and 2. **Self Study:**Master Linux commands, configure secure Kali Linux environment, and practice reconnaissance using footprinting and scanning tools.	**8**	**Apply (CO1)**
Module-2 Website and Network Security		
Website Penetration Testing: Installing OWASP**,** HTTP request**,** HTTP response**,** Burpsuite configuration**,** editing packets in Burpsuite**,** Whatweb and Dirb**,** Password recovery attack**,** Burpsuite login bruteforce**,** Hydra login bruteforce**,** Session fixation**,** Injection attacks**,** Finding blind command injection, SQL basics, Manual SQL injection part 1 and 2, SQLmap basics, XML injection, Installing XCAT and preventing injection attacks, Reflected XSS, Stored XSS, Changing HTML code with XSS, XSSer and XSSsniper.	**8**	**Apply (CO2)**

Simple command injection**,** Exploiting command injection vulnerability **Wireless Cracking:** Wireless attacking theory, Enabling monitor mode, Capturing handshake with airodump, RockYou.txt, Cracking with Aircrack, Cracking with Hashcat, Creating password lists with Crunch, Creating password lists with Cupp, Rainbow tables part 1 and 2, Installing Fluxion, Finding and cracking hidden networks, Preventing wireless attacks **Self Study:** Explore website penetration testing techniques, practice injection attacks, and learn wireless network cracking and password attacks.		
Module-3: Network Attacks and System Hacking		
Man in the Middle Attacks: Problems while installing MITMf, HTTP traffic sniffing, DNS spoofing and HTTPS password sniffing, Hooking browsers with BEEF, Screenshotting targets browser, Cloning any webpage, Ettercap basics.ARP protocol basics, Man in the middle attack theory, Installing MITMf, Manual ARP spoofing, **System Hacking with Metasploit:** MSFconsole environment, Metasploit modules explained, Bruteforcing SSH with Metasploit, Attacking Tomcat with Metasploit, Getting Meterpreter with command injection, PHP code injection, 2 Metasploitable exploits, Wine installation, Creating Windows payloads with Msfvenom, Encoders and Hex editor, Windows 10 Meterpreter shell, Meterpreter environment, Windows 10 privilege escalation, Preventing privilege escalation, Post exploitation modules, Getting Meterpreter over Internet with port forwarding, Eternalblue exploit, Persistence module, Hacking over the internet with Ngrok, Android device attack with Venom. **Self Study:** Understand man-in-the-middle attacks and master Metasploit framework for system exploitation and post-exploitation tasks.	**8**	**Analyze (CO3)**
Module-4: Exploitation and Scripting		
Coding an Advanced Backdoor: Theory behind reverse shells**,** Simple server code**,** Connection with reverse shell**,** Sending and receiving messages**,** Sending messages with while true loop**,** Executing commands on target system**,** Fixing backdoor bugs and adding functions**,** First test using our backdoor**,** Trying to connect every 20 seconds**,** Creating persistence part 1 and 2**,** Changing directory**,** Uploading and downloading files**,** Downloading files from the internet**,** Starting programs using our backdoor**,** Capturing screenshot on the target PC**,** Embedding backdoor in an image part 1 and 2, Checking for administrator privileges. **Python Crash Course:** Variables, Raw input, If-else statements, for loops, while loops, Python lists, Functions, Classes, importing libraries, Files in Python, Try and except rule. **Self Study:** Learn to code, test, and enhance advanced reverse shell backdoors with persistence, file operations, and privilege management.	**8**	**Apply (CO4)**
Module-5: Malware Development and Career Guidance		
Creating a Keylogger for the Backdoor: Adding report function**,** writing keystrokes to a file**,** Adding the keylogger to our reverse shell part 1 and 2**.** **Basic Authentication Brute forcer:** Printing banner, adding available options, starting threads for brute force, writing function to run the attack, Brute forcing router login, Bypassing antivirus with your future programs, Sending malware with spoofed email, Importing Pynput**,** Simple keylogger **BONUS - Building a Career in Cyber Security** **Self Study:** Develop keyloggers and brute force tools, practice malware delivery techniques, and explore career paths in cybersecurity.	**8**	**Apply (CO5)**

Course outcomes:

The students will be able to

1. Applying fundamental ethical hacking concepts, including secure setup, Linux commands, and principles.
2. Construct effective foot printing and scanning techniques to identify vulnerabilities and gather critical information.
3. Analyze and develop skills in website penetration testing and wireless security cracking to assess and exploit vulnerabilities in web applications and networks.
4. Build and utilize advanced tools for system exploitation, including Metasploit, and develop scripting and malware techniques to enhance ethical hacking skills.
5. Implement advanced security tools like backdoors and keylogger, applying these skills to real-world scenarios while understanding ethical implications and career opportunities.

Reference Books:

https://www.udemy.com/course/ethical-hacking-kali-linux/?couponCode=IND21PM

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VII

R Programming (PE-IV MOOC)
Credit: 3

Course Code	23AML17354	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

- Learn R programming language, data science. data types, vectors, Matrices and Functions.
- Apply and analyze the problems using Intermediate R & Data Manipulation techniques.
- Study the Data Visualization in R by plotting, mapping, reports creation.
- Introduction to Machine Learning, Linear Regression techniques and Data Processing.
- Understanding Linear and Logistic Regression and applying R program in real life.

Prerequisites: Basic knowledge of programming

Modules	Number of Hours	Bloom's Level
Module-1: Introduction, Getting Started with R, Data Types and Structure in R		
Data Science and Machine Learning Course Intro: Intro, what is Data Science? ML Overview, Data science + Machine Learning Marketplace, DS and ML Job Opportunities. **Getting Started with R:** started with R, R basics, working with Files, R Studio, Tidyverse Overview, Additional Resources. **Data Types and Structure in R:** overview, Basic Types, Vector part 1 &2,vector missing values, Coerction, Naming, Misc. ,working with matrices, working with Lists, Introduction to Data Frames, Creating Data Frames, Data frames: Helper Functions, Tibbles.	**8**	**(CO1) Apply**
Module-2: Intermediate R, Data Manipulation in R		
Intermediate R: Intro, Relational Operators, Logical, Operators, Conditional Statements, working with Loops, Functions, Packages, Factors, Data & Times, Functional Programming, Data Import/Export, Working with Databases. **Data Manipulation in R**: Intro, Tidy Data, The Pipe Operator,{dplyr}:The Filter Verb, Mutate verb, Arrange Verb, Summarize Verb	**8**	**(CO2) Apply**
Module-3: Data Manipulation in R, Data Visualization in R		

Data Manipulation in R:continue: Data Pivoting:{tidyr},String Manipulation;{stringr},Web Scraping:{rvest},JSON Parsing:{jsonlite}. **Data Visualization in R:** Introduction, getting started with Data Visualization in R, Aesthetics Mappings, Single variable Plots, Two variable Plots, Facets, Layering and Cordinate systems, Styling and Saving. **Creating Reports with R Markdown:** Introduction to R Markdown.	**8**	**(CO3) Analyze**
Module-4: Introduction to Machine Learning, Data Preprocessing, Linear Regression		
Building webapps with R Shiny: Introduction to R Shiny, Creating A basic R Shiny App, Other Examples with R Shiny. **Introduction to Machine Learning:** Introduction to ML part 1 & part 2 **Data Preprocessing:** Data Processing Introduction, Explanation -1hr **Linear Regression: A simple Model:** LR: Simple Model Introduction, Explanation	**8**	**(CO4) Apply**
Module-5: Linear Regression, Logistics Regression		
Exploratory Data Analysis: Introduction, Hands-on Exploratory Data Analysis **Linear Regression – Real Model:** Introduction, LR Real Model **Logistics Regression:** Introduction, Logistic Regression in R **Starting A career in Data Science:** Starring a Data Science Career Overview, creating a Data Science Resume, getting started with Freelancing, Top Freelancing Websites, Personal Branding, Networking Do's and Don'ts, Setting Up a website.	**8**	**(CO5) Analyze**

Course outcomes:

1. Understand and apply the R Programming concepts, data types, functions, Packages and work with Database.
2. Experiment with Data Manipulation techniques using Dplyr and Tidyr .
3. Study and analyze Data Visualization techniques, mapping, plotting and creating Reports
4. Build webapps and make use of Machine Learning and Linear Regression model and Data Preprocessing.
5. Apply Logistic Regression and analyze data for real world problem solving.

Reference Books:

1. https://www.udemy.com/course/data-science-and-machine-learning-with-r-from-a-z/?couponCode=OF83024F
2. Data Science and Machine Learning with R by Reema Thareja, Mc Graw Hill,1st Edition 2021
3. https://bookdown.org/rdpeng/rprogdatascience/
4. MachineLearningR__Brett_Lantz.pdf by research gate network.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VII

Research Methodology & Intellectual Property Rights (AEC)
Credit: 2

Course Code	23AML174	CIA Marks	50
Teaching Hours/Week (L: T: P: J)	2:0:0:0	SEA Marks	50
Total Number of Lecture Hours	25	Exam Hours	03

Course Learning Objectives:

This course will enable students to

- Understand the basics of research methodology and apply Ethics in conducting research work
- Analyze the concept of Literature Review, Technical Reading and writing, Attributions and Citations while preparing a technical report
- Learn the concepts of Intellectual Property Rights and analyze and apply methods of filing and publishing a patent.
- Analyze the knowledge of different categories of Intellectual Property Rights for copyrights, trademarks, Industrial Designs and Geographical Indication

	No. of Hours	**Bloom's Level**
Module-1		
Introduction: Meaning of Research, Objectives of Engineering Research, and Motivation in Engineering Research, Types of Engineering Research, Finding and Solving a Worthwhile Problem. **Ethics:** Ethics in Engineering Research Practice, Types of Research Misconduct, Ethical Issues Related to Authorship. Self study: Key Principles of Engineering Ethics in Research	**5**	**(CO1) Apply**
Module-2		
Literature Review and Technical Reading: New and Existing Knowledge, Bibliographic Databases --- Web of Science, Google Scholar, Introduction to Technical Reading, Taking notes. **Attributions and Citations:** Citations: Functions and Attributes, Impact of Title and Keywords on Citations, Knowledge Flow through Citation -- Citing Datasets, Styles for Citations, Acknowledgments and Attributions, What Should Be Acknowledged, Acknowledgments in Books, Dissertations.	**5**	**(CO2) Analyze**

Technical Writing and Publishing: Attributes and Reasons of Technical Writing, Patent or Technical Paper? Writing Strategies, Journal Paper: Structure and Approach, Publish Articles to Get Cited or Perish. Self study: Book chapter writing		
Module-3		
Introduction To Intellectual Property: Role of IP in the Economic and Cultural Development of the Society, IP as a Global Indicator of Innovation, Major Amendments in IP Laws and Acts in India. **Building Intellectual Property Rights:** Codes and Standards, Patents: Subject and Importance, Requirements for Patentability, Application Preparation and Filing, Patents and Commercialization in Academia. **Case Studies on Patents:** Case study of Curcuma (Turmeric) Patent, Case study of Neem Patent, Case study of Basmati patent. Self study: Other related patent study	**5**	**(CO3) Analyze**
Module-4		
Copyrights and Related Rights: Classes of Copyrights. Criteria for Copyright. Ownership of Copyright. Copyrights of the Author. Copyright Infringements. Fair Use Doctrine. Copyrights and Internet. Non-Copyright Work. Copyright Registration. Interesting Copyrights Cases. **Trademarks:** Eligibility Criteria. Who Can Apply for a Trademark. Acts and Laws. Designation of Trademark Symbols. Classification of Trademarks. Registration of a Trademark is Not Compulsory. Validity of Trademark. Trademark Registry. Process for Trademarks Registration -- Prior Art Search. Famous Case Law: Coca-Cola Company vs. Bisleri International Pvt. Ltd. Self study: Related case studies on Copyrights	**5**	**(CO4) Analyze**
Module-5		

Industrial Designs: Eligibility Criteria. Acts and Laws to Govern Industrial Designs. Design Rights. Enforcement of Design Rights. Non-Protectable Industrial Designs India. Protection Term. Procedure for Registration of Industrial Designs -- Prior Art Search, Application for Registration. Application Forms. Classification of Industrial Designs. Famous Case Law: Apple Inc. vs. Samsung Electronics Co. **Geographical Indications:** Acts, Laws and Rules Pertaining to GI. Ownership of GI. Rights Granted to the Holders. Registered GI in India. Identification of Registered GI. Classes of GI. Non-Registerable GI. Protection of GI. Procedure for GI Registration Documents Required for GI Registration. Famous Case Law: Banglar Rasogolla Vs. Odisha Rasogolla Self study: Famous GI examples	**5**	**(CO4) Analyze**

Course outcomes:

1. Understand types of research methodologies and apply Ethics in conducting research work
2. Examine and analyze Literature Review, Technical Reading & writing, Attributions, and Citations while preparing a technical report
3. Analyze the significance of Intellectual Property Rights, patentability requirements, and patent commercialization strategies in fostering innovation and technological advancement.
4. Differentiate and analyze various forms of intellectual property protection, including copyrights and trademarks, with respect to ownership, infringement, registration procedures, and legal implications.
5. Demonstrate the ability to prepare research papers, patents and copyright following ethical and academic standards.

Reference Books:

1. Dipankar Deb, Rajeeb Dey, Valentina E. Balas “Engineering Research Methodology”, ISSN 1868-4394 ISSN 1868-4408 (electronic), Intelligent Systems Reference Library, ISBN 978-981-13- 2946-3 ISBN 978-981-13-2947-0 (eBook), https://doi.org/10.1007/978-981-13-2947-0

2. Intellectual property A primer for Academia, Prof. Rupinder Tewari, Ms. Mamta Bhardwaj, Publication Bureau, Panjab University, ISBN: 81-85322-92-9, Edition: 2021.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

Reverse Engineering & Malware Analysis (PE-V MOOC)

Credit - 3

SEMESTER– VIII

Course Code	23AML18145	CIE Marks:	50
Teaching Hours/Week (L: T: P: J)	3:0:0:0	SEE Marks:	50
Total Number of Lecture Hours	40	SEE Duration:	3 Hours

Course Learning Objectives:

This course will enable students to

- Introduced with the tools and techniques used in reverse engineering and malware analysis, specifically focusing on Ghidra, binary analysis, and file structures.
- Provide foundational knowledge of Linux, its terminal, file structure, and commands necessary for malware analysis and reverse engineering.
- Enhance students' understanding of C++ concepts and programming, particularly in relation to reverse engineering and malware analysis.
- Develop proficiency in handling pointers, functions, and object-oriented concepts in C++ for reverse engineering tasks.
- Provide a deep dive into Abstract Data Types (ADT) and node-based structures in C++ for analyzing and manipulating complex data types in malware analysis.

Prerequisites: Basic knowledge of programming, cybersecurity and OS

Module 1: Introduction	**Number of Hours**	**Bloom's Level**
Introduction: Downloading Ghidra and File Structure, Installing JDK in Linux, Installing JDK in Windows or MacOS, opening project in Ghidra, Discovering Ghidra Malware Analysis and Reverse Engineering with Ghidra, **Binary Analysis:** Analysis of Binary and 4 Stages of Compilation, Preprocessing, Compilation Phase, Assembly Phase, Linking Phase Symbols, Stripped and Not Stripped Binaries. Linux - ELF Format, Windows - PE Format, Learning the Windows PE Format, IDA Pro / IDA Freeware, Linux for Reverse Engineering and Malware Analysis.	**08**	**(CO1) Apply**
Module 2: Starting with Linux Terminal		
Starting with Linux Terminal: Understanding Key Concepts, Finding Helpful Manuals, Understanding Linux Directories, Linux Directories - Part 2, Starting with Linux shell and Exploring Linux Systems, **Permissions, Managing Processes and Packages in Linux:** Fundamentals of Processes in System, The Magic with PS Aux Command, The Magic with TOP Command, Foreground and Background Processes, From Background to Foreground, The Tutorial with Linux Process Signals, Introduction to Packet Management Systems, Why we need package management tools, Package Management in Linux - Final, Trcrt command, Networking with Linux.	**08**	**(CO2) Apply**
Module 3: C++ for Reverse Engineering:		
C++ for Reverse Engineering: CPP Newlines, Tabs and Spaces, Initializers in CPP, CPP Writing Styles, Statements and Expressions, RVALUE and LVALUE, Comma, Types and Variables, Literals in CPP, Defining Constants, Const Expressions, Beginning to Pointers, Namespaces in CPP, Scoping Variables and Static Keyword,	**08**	**(CO3) Apply**

Conditional Statements, Variable Types in C/C++, Reverse Engineering and C: Introduction to CPP Classes, Pointers and Arrays, Pointers and Arrays		
Module 4: Pointers and Functions/ Reverse Engineering and Malware Analysis		
Pointers in Practice, Pointers in Practice - Part 2, Pointers in Practice - Part 3, Pointers in Practice - Part 4, Introduction to Functions, Functions - Part 1, Functions - Part 2, Functions - Part 3, Functions - Part 4, Functions - Part 5, Functions - Part 6, Functions - Part 7, Introduction to Basic CPP Programming, Creating Random Card Generator Using Enum and rand, Developing Main Function and Using static_cast, Introduction to ADT, Creating ADT Class and Using Constructors, Virtual Methods and Overriding from Class, Controlling Copy Operations with ADT, Starting with Templates, Class Templates in CPP, Introduction to STL, Algorithm Analysis, Scenario Analysis, Developing Main, Factorial Method.	**08**	**(CO4) Apply**
Module 5: ADT and nodes in C++ for Reverse Engineering		
Creating List ADT: Developing Get and Insert Methods, Developing Search and Remove Methods, Testing ADT List, Node and Chaining Example, Creating and Chaining Nodes with Different Data Types, Creating Custom Linked List, Get Method, Developing InsertHead, InsertTail and Insert Methods, Developing Search Operation, Removing Items. **Arrays for Reverse Engineering and Malware Analysis:** Understanding Arrays, Manipulating Arrays, Starting with Array Pointers, Pointer Increment vs Array Indexing.	**08**	**(CO5) Apply**

Course Outcomes:

At the end of the course, the students will be able to:

1. Understand and Apply Ghidra for malware analysis and reverse engineering tasks, including binary analysis and file structure identification.
2. Utilize Linux terminal effectively for system exploration, process management, and networking tasks related to reverse engineering.
3. Design and debug C++ programs, including understanding key concepts such as pointers, arrays, functions, and classes, for reverse engineering purposes.
4. Apply advanced C++ techniques such as ADTs, templates, and object-oriented programming to solve reverse engineering problems.
5. Develop and manipulate custom data structures like linked lists and arrays, and understand how they can be used in reverse engineering and malware analysis scenarios.

References

1. https://www.udemy.com/course/reverse_engineering/?srsltid=AfmBOoolqHTZDPJuRs62cQwkBq7Te6zw6vVkjq6PfYv3jL9xXeEx2CFk&couponCode=NVDIN35
2. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software Michael Sikorski, Andrew Honig No Starch Press, 1 Feb 2012.
3. Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware Paperback by Monnappa K a, Packt Publishing, 29 June 2018.
4. Malware Analysis and Detection Engineering: A Comprehensive Approach to Detect and Analyze Modern Malware Paperback, by Abhijit Mohanta, Anoop Saldanha, 23 September 2020.

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Artificial Intelligence and Machine Learning

SEMESTER – VIII

Graph Neural Networks (PE-V MOOC)
Credit: 3

Course Code	23AML18155	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

- Understand the fundamental concepts of Graph Theory and their application in Machine Learning.
- Learn about different types of Graph Neural Networks (GNNs) and their use cases.
- Gain knowledge about using GNNs and ANN for solving real world problems.
- Deep understanding of advanced GNN optimization models in various GNN Applications

Prerequisites: Fundamentals of Graph Theory and CNN.

	Number of Hours	**Bloom's Level**
Module-1 Introduction to Graphs & Traditional Graph Algorithms		
Overview of Graphs: Definition of graphs- nodes, edges, adjacency matrix, and graph representations, Types of graphs- directed vs. undirected, weighted vs. unweighted, bipartite graphs, Implementation of different graphs using TensorFlow **Graph Traversal Algorithms**-Depth-First Search (DFS), Breadth-First Search (BFS), Implementation of DFS and BFS using tensorflow **Graph Shortest Path Algorithms**-Dijkstra's Algorithm, Bellman-Ford, Implementation using tensorflow **Self Study: Components:** Implementation of different graphs, DFS, BFS, Dijkstra's , Bellman- Ford algorithms using Tensorflow.	**8**	**(CO1) Apply**
Module-2: Neural Network & Graph Neural Networks		
Convolutional Neural Networks (CNNs)-Basics of CNNs: convolutions, pooling, and applications in image processing. **Introduction to Graph Neural Networks**-What is a Graph Neural Network? Motivations and benefits, Comparing GNNs to traditional neural networks and CNNs. **Message Passing Framework**-Explanation of the message passing mechanism: aggregation, update, and combination, Neighborhood aggregation and node embeddings. **Self Study Components:** The GNN and Message Passing frameworks implementation using Tensorflow.	**8**	**(CO2) Apply**
Module-3: Key GNN Models		

Graph Convolutional Networks (GCN)- Architecture and layers of GCNs, Mathematical formulation and operation, GCN for node classification. **Graph SAGE (Graph Sample and Aggregation)** - Localized Neighborhood aggregation, Sampling techniques and their importance for scalability. **Graph Attention Networks (GAT)**- Self-attention mechanism in graphs, Node-wise attention in the aggregation process. **Self Study Components:** Implementation of the **Graph Attention Network (GAT)** using **TensorFlow** with the **Spektral** library	**8**	**(CO3) Apply**
Module-4: Advanced Topics in GNNs		
Graph Pooling Techniques-Pooling methods: Global Average Pooling, Top-K Pooling, DiffPool, The role of pooling in graph classification tasks. **Graph Autoencoders**-GNNs for unsupervised learning, Applications in graph reconstruction and anomaly detection. **Graph Generative Models**-Generative approaches for graph creation and data synthesis, Variational Graph Autoencoders (VGAE). **Self Study Components:** Implementation of **Variational Graph Autoencoder (VGAE)** using **PyTorch** with the **PyTorch Geometric** library	**8**	**(CO4) Apply**
Module-5: Optimization and Applications of GNNs		
Optimization Techniques for GNNs-Gradient descent and variants: Adam, RMSProp. **Node Classification Application**- Use cases in social networks, recommendation systems. **Link Prediction Application**-Predicting edges in a graph for applications like social network growth or fraud detection. **Self Study Components:** Implement and optimize various GNN applications using Adam and RMSprop optimizers in PyTorch/Tensorflow.	**8**	**(CO5) Apply**

Course outcomes:

The students will be able to

1. Apply graph traversal and shortest path algorithms in Machine Learning Applications.
2. Explore different types of Graph Neural Networks (GNNs) and their use cases.
3. Apply ANN architectures for Graph reconstructions and anomaly detection
4. Apply advanced GNN models for solving problems in fields such as social networks, recommendation systems, and bioinformatics.
5. Analyze GNN Optimization techniques in various GNN applications.

Reference Books:

1. *Graph Representation Learning* by William L. Hamilton
2. *Deep Learning on Graphs* by Yao Ma and Jure Leskovec
3. *Machine Learning with Graphs* by Xavier Bresson and Thomas Laurent