SEMESTER I

PRINCIPLES OF PROGRAMMING WITH C

Core Course - II	24UCP401	Principles of Programming with C	Credits - 4

Course Outcomes: At the end of the course the student will be able to

CO1	Elucidate the basic architecture and functionalities of a Computer
CO2	Understand the programming paradigm of C programming languages
CO3	Implement the concept of arrays, strings and functions
CO4	Apply code reusability with functions and pointers
CO5	Develop C programs using structure, union and file concepts

SYLLABUS

UNIT I: Principles of Computer & Overview of C

Introduction - Components of a Computer – Concept of Hardware and Software – Art of Programming through Algorithms and Flowcharts - History of C – Importance of C – Sample Programs – Basic Structure of C programs– Executing C Program.

UNIT II: Constants, Decision Making and Looping

Introduction – Character set – C Tokens - Constants, Variables and Data Types – Declaration of Variables - Declaration of Storage Class – Symbolic Constants – Operators and Expressions –Reading and Writing a Character – Formatted Input – Formatted Output - Decision Making and Branching – Decision Making and Looping.

UNIT III: Arrays & Functions

Arrays – one dimensional array – Declaration & Initialization and access- 2D arrays and multidimensional arrays- Strings – defining strings, initializing, accessing String handling functions- Functions Introduction – Elements of User – defined Functions – Definition of Functions – Function Declaration – Category of Functions – Nesting of functions - Recursion – Passing Arrays to Functions.

UNIT IV: Structure & Union

Structures: Introduction – Defining a structure – Declaration of structure – Accessing Structures members – Array of Structures – Arrays within Structures - Structures within structures – Structures and functions – Union.

UNIT – V: Pointers & Files

Introduction – Declaring pointer variable – Initialization of Pointer variables - Chain of Pointers - Pointer Expressions —Pointers and Arrays – Array of Pointers – Pointers as function arguments – Functions returning Pointers – Pointers to Functions – Pointers and Structures - Files: Introduction – Defining and Opening a File – Closing a File – Input / Output operations on files.

Text Books:

1. E. Balagurusamy, “Programming in ANSI C”, McGraw Hill, 8th Edition, 1st January 2024.

Reference Books:

1. Jyoti P Mirani & Vikas B. Agarwal, “Computer Fundamentals”, 2nd Edition, Nirali Prakashan, 2019.

2. Yashavant Kanetkar, “Let Us C”, 18th Edition, BPB Publications, 2021.
3. Szuhay, Jeff, and Szuhay, Jeff, “Learn C Programming: A Beginner's Guide to Learning C Programming the Easy and Disciplined Way”, Packt Publishing, 2020.
4. Jena, Sisir Kumar, and Jena, Sisir Kumar, “C Programming: Learn to Code”, 1st Edition, CRC Press, 2021.
5. Reema Thareja , “Programming in C”, 3rd Edition, Oxford University Press, 2023.

PRINCIPLES OF PROGRAMMING WITH CLAB

Course Outcomes: At the end of the course the student will be able to

CO1	Remember and understand the logic of given concept and construct the programs
CO2	Apply the concept of single and multi-dimensional arrays to solve problems related to searching, sorting and matrix operations
CO3	Apply the concept of Strings for writing programs related to character array.
CO4	Write programs using concept of user defined and recursive functions.
CO5	Make use of different data-structures like arrays, pointers, structures and files

SYLLABUS

1. Write C program
   (i) To convert temperature from degree Centigrade to Fahrenheit,
   (ii) Find whether given number is even or odd,
   (iii) Find the greatest of three numbers.
2. Write a C program to find the sum, average, standard deviation for a given set of numbers.
3. Write a C program to print magic square of order n where n>3 and n is odd.
4. Write C program to solve quadratic equation using functions.
5. Write C program
   (i) To swap two numbers using call by value
   (ii) To change the elements of an array using call by reference
6. Write C program to count the number of vowels in the given sentence.
7. Write C program to find the factorial of a given number using a recursive function.
8. Write C program to reverse a number using pointer.
9. Write C program to check whether the given string is a palindrome using pointers.
10. Write C program to print the students mark sheet assuming roll no, name, and marks in 5 subjects in a structure. Create an array of structures and print the mark sheet in the university pattern.
11. Write a function using pointers to add two matrices and to return the resultant matrix to the calling Function.
12. Write a C program which receives two file names as arguments and check whether the file contents are same or not. If same delete the second file.

DIGITAL COMPUTER FUNDAMENTALS

Core Course - I	**24UCP402**	**Digital Computer Fundamentals**	**Credits - 4**

COURSE OUTCOMES: At the end of the course the student will be able to

CO1	Understand various types of number systems and their conversions
CO2	Familiar with the Boolean algebra, logic gates, logical variables and its the truth table
CO3	Design and implement variety of logical devices using combinational circuits concepts.
CO4	Analyze sequential circuits like Registers and Counters using flip-flops
CO5	Demonstrate and compare the construction of programmable logic devices and different types of ROM

SYLLABUS

UNIT I: Number Systems

Digital Systems– Binary numbers – Number base conversions- Octal, Hexadecimal -one's complement and two's complement, Binary Arithmetic, Fixed point representation and Floating point representation of– Signed Binary Numbers - Binary Codes

UNIT II: Boolean Algebra & Logic Gates

Boolean Algebra –Basic definitions – Axiomatic Definition of Boolean Algebra -Basic Theorems and Properties of Boolean Algebra – Boolean Functions- Canonical and standard forms –The AND gate, The OR gate, The NAND gate, NOR gate, The Exclusive–OR gate and Exclusive-NOR gate- Boolean Algebra and Logic Simplification- DeMorgan's Theorems, Boolean Expressions and Truth Tables, The Karnaugh Map, SOP, POS, Don't care conditions.

UNIT – III: Combinational Logic Circuits

Basic combinational Logic Circuits, Implementing Combinational Logic, The Universal Property of NAND and NOR Gates. Functions of Combinational Logic - Adders, Subtractors, Decoders, Encoders, Code Converters, Multiplexers, Demultiplexers.

UNIT – IV: Sequential Circuit Design

Flip flops (RS, Clocked RS, D, JK, JK Master Slave, T, Counters, Shift registers and their types, Counters: Synchronous and Asynchronous counters.

UNIT – V: Memory and Programmable Logic

Introduction, Random access memory, memory decoding, error detection and correction, read only memory, Flash Memory - Special Types of Memories, Magnetic and Optical Storage- programmable logic array, sequential programmable devices.

Text Book :

1. M. Morris Mano, Michael D. Ciletti, "Digital Design", 6th Edition, Pearson Education, 2018.

Reference Book :

1. Donald P leach, Albert Paul Malvino, Goutam Saha, Digital Principles and Application, 8th Edition., McGraw Hill education (India) Private Limited, 2015.
2. Prof. Satish Jain Shashi Singh, "Digital Computer Fundamentals", 1st Edition, BPB Publications, 2017.
3. Ronald Tocci, Neal Widmer , Gregory Moss," Digital Electronics Principles and applications,12th Edition, Pearson, 2017.
4. Digital Electronics Circuits and Systems, Bishnu Charan Sarkar and Suvra Sarkar,1st Edition, Sad-Ul-Tan Prakasani, Burdwan, 2018.
5. Roger L. Tokheim, Patrick E., "Digital Electronics: Principles and Applications", 9th Edition, McGraw-Hill Education, 2022.

SEMESTER II

INTRODUCTION TO ARTIFICIAL INTELLIGENCE
COURSE CODE: 24UAM501

SYLLABUS

UNIT I: Introduction: Introduction- What is AI – The Foundations of Artificial Intelligence – The History of the Artificial Intelligence – Risks and benefits of AI – Intelligent Agents: Agents and Environments – Good Behavior: The Concept of Rationality – The Nature of environments – The Structure of Agents.

UNIT II: Problem-Solving: Solving Problems by Searching: Problem-solving Agents – Examples Problems – Search Algorithms – Uninformed Search Strategies – Informed (Heuristic) Search Strategies – Heuristic Functions – **Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems – Constraint Propagation: Inference in CSPs – Backtracking Search for CSPs – Local Search for CSPs – Structure of Problems.

UNIT III: Knowledge, reasoning and planning: Logical Agents: Knowledge-based Agents – Logic – Propositional Logic: A Very Simple Logic – Propositional Theorem Proving – Effective Propositional Model Checking – Agents based on Propositional Logic – **First-Order Logic:** Representation Revised – Syntax and Semantics of First-Order Logic – Using First-Order Logic - Knowledge Engineering in First-Order Logic.

Unit IV: Inference in First-Order Logic: Propositional VS First-Order Inference – Unification and First-Order Inference – Forward Chaining – Backward Chaining – **Knowledge Representation:** Ontological Engineering – Categories and Objects – Events – Metal Objects and Modal Logic – Reasoning Systems for Categories – Reasoning with Default Information.

UNIT V: Uncertain Knowledge and Reasoning: Quantifying Uncertainty: Acting under Uncertainty – Basic Probability Notation – Inference using Full Joint Distribution – Independence – Bayes' Rule and its use – Naïve Bayes Models – **Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain – The Semantics of Bayesian Networks.

ARTIFICIAL INTELLIGENCE LAB
COURSE CODE: 24UAM551

SYLLABUS

1. Write a program to create AI-powered Chatbots application
2. Write a program to create a Simple Rational Agent
3. Write a program to implement Uninformed Search Technique: Depth First Search
4. Write a program to implement A* Algorithm
5. Write a program to implement t Naïve Bayes Models
6. Write a program to solve constraint satisfaction problems.
7. Write a program to implement Tic-Tac-Toe game using
8. Write a program to implement Game Playing Algorithms: Alpha Beta Purning

SEMESTER III

Core Course – III
24UCP403: Python Programming

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit - I	CO1	Understand the basic building blocks of programming constructs	K1, K2
Unit - II	CO2	Apply control structures such as loops, conditionals, and iterators to solve basic computational problems.	K1 – K4
Unit - III	CO3	Develop modular Python programs using functions, handle files and exceptions effectively, and use standard libraries.	K1 – K4
Unit - IV	CO4	Implement object-oriented programming concepts such as classes, objects, methods, encapsulation, polymorphism, and operator overloading.	K1 – K4
Unit - V	CO5	Analyze and work with scientific computing libraries like NumPy and pandas to perform efficient data manipulation and numerical operations.	K1 – K4

Syllabus

Unit I

Introducing Python - Python on Different Operating Systems: Python on Windows- Python on macOS - Python on Linux - Variables and Assignment Statements – Arithmetic – Function print and an Introduction to Strings.

Unit II

Decision Making: The if Statement and Comparison Operators - Objects and Dynamic Typing-Control Statements: If Statement-If Else and If Elif Else Statements -While Statement, For Statement -Iterables, Lists and Iterators -Builtin Range Function -Augmented Assignments –Sequences: Lists and Tuples-Dictionaries and Set-controlled Iteration; Formatted Strings –Sentinel controlled Iteration - Builtin Function Range-Break and Continue Statements.

Unit III

Functions: Defining Functions-Functions with Multiple Parameters- Python Standard Library -math Module Functions - Using IPython Tab Completion for Discovery - Default Parameter Values -Keyword Arguments -Arbitrary Argument Lists- Methods: Functions That Belong to Objects- Scope rules- Recursion- Files: Text File Processing - Updating Text Files - Handling Exceptions- finally Clause -Explicitly Raising an Exception.

Unit IV

Object Orientation: Custom Class Account- Definition-Composition: Object References as Members of Classes- Controlling Access to Attributes-Properties for Data Access- Account Class Definition-Composition: Object References as Members of Classes -Controlling Access to Attributes-Properties for Data Access- Duck Typing and Polymorphism- Operator Overloading- Namespaces and Scopes.

Unit V

NumPy Basics: Arrays and Vectorized Computation: The NumPy ndarray: A Multidimensional Array Object- Pseudorandom Number Generation- Universal Functions: Fast Element-Wise Array Functions- Array-Oriented Programming with Arrays- File Input and Output with Arrays- Linear Algebra- Getting Started with pandas: Introduction to pandas Data Structures- Essential Functionality.

Text Book:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021(Unit-I-IV)
2. Python for Data Analysis Data Wrangling with pandas, NumPy, and Jupyter, Wes McKinney, oreilly ,3rd Edition,2022(Unit-V)

Reference Books:

1. Python Crash Course by Eric Matthes, 3rd Edition, No Starch Press, January 2023
2. Martin C. Brown, Python: The Complete Reference, McGraw Hill Education (India), 2024.
3. Jeeva Jose and P. SojanLal, “Introduction to Computing and Problem Solving with PYTHON”, Khanna Book Publishing Co, 2016.
4. Phill Vega., "Python Programming 2020: Guide to Learn Python, Step by Step, with Practical Exercises, Tips and Tricks”, Independently Published, 2020.
5. Reema Thareja,"Python Programming: Using Problem Solving Approach”, Oxford University Press, 2019.
6. Kanetkar Aditya Yashavant, Kanetkar, Let Us Python, 2nd Edition, BPB Publications, 2020.

Core Laboratory – III

24UCP452: Python Programming Lab

Credits: 2 **Internal Marks: 40**

Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
CO1	Understand and explain Python syntax, data types, operators, and control structures.	K3
CO2	Apply functions, loops, and conditional statements to develop modular and logical Python programs.	K3
CO3	Analyze problems and implement appropriate data structures such as lists, dictionaries, and classes.	K4
CO4	Develop Python applications using object-oriented concepts and file handling techniques.	K4
CO5	Implement algorithms for sorting, searching, and graph traversal and demonstrate their functionality.	K3,K4

Syllabus

1. Write a Python program to simulate an ATM using conditional statements.
2. Write a Python program for a BMI calculator with health condition output.
3. Write a Python program to count word frequency in a sentence.
4. Write a Python program to compute the factorial of a number using both loops and recursion.
5. Write a Python program for an inventory management system using dictionary.
6. Write a Python program for an employee attendance tracker using dictionary.
7. Write a Python program to add two matrices of the same size and display the result.
8. Write a Python program to find the transpose of a matrix.
9. Write a Python program for a bank transaction system using functions.
10. Write a Python program for a to-do list application using file handling.
11. Create a program to count the number of times a specific word appears in a given text file.
12. Implement a BankAccount class with methods to deposit, withdraw, and display the balance.
13. Create a Vehicle class and derive Car and Bike subclasses from it.
14. Write a Python program Basic Array Operations using NumPy
15. Write a Python program to create and display a DataFrame using Pandas.
16. Write a Python program to implement a Stack and queue data structure using a list. Include push, pop, peek, and is_empty operations.
17. Write a Python program to implement a singly linked list with the following operations:
    Insertion at the end
    Deletion of a node by value
    Display the list
18. Write a Python program to implement the following sorting algorithm using Bubble Sort.
19. Write a Python program to implement Linear Search and Binary Search.
20. Write a Python program to implement graph traversal using Breadth-First Search (BFS) and Depth-First Search (DFS).

Core Course - V
24UCP404: Data Structures and Algorithms

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No.	CO. No.	Upon completion of this course, the students will be able to	Cognitive Level
Unit - I	CO1	Explain the concept of algorithms and analyze their performance using asymptotic notations.	K1 – K4
Unit - II	CO2	Implement and use linear data structures (arrays, stacks, queues, linked lists).	K1 – K4
Unit - III	CO3	Apply tree and graph-based data structures for hierarchical and networked data modelling.	K1 – K4
Unit - IV	CO4	Design and implement efficient sorting and searching algorithms using Divide and Conquer.	K1 – K4
Unit - V	CO5	Solve optimization problems using Greedy and Backtracking techniques.	K1 – K4

Syllabus

Unit I

Introduction of Algorithms, Analysing Algorithms: Theoretical Foundation, Algorithms and it's Specification, Random Access Machine Model ,Counting Primitive Operations, Notion of best case, average case and worst case, Characterizing Run Time, Use of asymptotic notation, Big-Oh Notation, Little-Oh, Omega and Theta Notations.

UNIT II

Arrays: Sparse Matrices–Representation of Arrays- Stacks and Queues - Fundamentals–Evaluation of Expression Infix to Postfix Conversion - Multiple Stacks and Queues - Linked List: Singly Linked List - Linked Stacks and Queues – Polynomial Addition- More on Linked Lists – Sparse Matrices – Doubly Linked List and Dynamic – Storage Management –Garbage Collection and Compaction

UNIT III

Basic Terminology – Binary Trees – Binary Tree Representations – Binary Trees-Traversal-More on Binary Trees – Threaded Binary Trees – Binary Tree.Representation of Trees – Counting Binary Trees. Graphs: Terminology and Representations-Traversals, Connected Components and Spanning Trees, Shortest Paths and Transitive Closure.

UNIT – IV

Introduction to Algorithms - Priority Queues - Heaps – Heap Sort – Divide and Conquer Algorithms - Binary Search - Merge Sort – Quick Sort — Finding the Maximum and Minimum.

UNIT - V

Greedy Method: The General Method – Knapsack Problem – Minimum Cost Spanning Trees– Optimal Merge Patterns-Single-Source Shortest Paths. Back Tracking: The General Method – The 8-Queens Problem – Sum of Subsets – Graph Coloring – Hamiltonian Cycles.

Text Book(s):

1. Ellis Horowiz, SartajSahni, "Fundamentals of Data Structure",Galgotia Publications(UNIT I-III)
2. Ellis Horowiz, SartajSahni and SanguthevarRajasekaran, "Computer Algorithms", University Press. (UNIT IV&V)

Reference Book(s):

3. Seymour Lipschutz, "Data Structures",TataMcgraw Hill, Schaum's Outline Series, 2014.
4. Micheal T. Goodrich and Roberto Tamassia: Algorithm Design: Foundations, Analysis and Internet examples (John Wiley & Sons, Inc., 2006).
5. ThomasH.Cormen,ChalesE.Leiserson,RonaldL.Rivest,CliffordStein,–Introduction to Algorithms‖,3rdEdition,McGrawHill, 2010.

SEMESTER IV

Core Course – VI
24UAM502: Introduction to Machine Learning

Syllabus

UNIT I

Introduction: What is machine learning? - Examples of Machine Learning Applications lHistory - Supervised Learning - Learning a class from Examples –Vapnik-Chervonenkis Dimension - Probably Approximately Correct Learning – Noise - Learning Multiple Classes – Regression Model Selection and Generalization - Dimensions of a Supervised Machine Learning Algorithm.

UNIT II

Bayesian Decision Theory: Introduction – Classification - Loses and Risks - Discriminant Functions - Association Rules - Parametric Methods: Introduction –Maximum Likelihood Estimation-Evaluating an Estimator: Bias and Variance-The Bayes Estimator-Parametric Classification-Regression-Tuning Model Complexity: Bias/Variance delimma - Model Selection Procedures.

UNIT III

Multivariate Methods: Multivariate data - Parameter Estimation - Estimation of Missing Values-Multivariate Normal Distribution - Multivariate Classification - Tuning Complexity Discrete Features - Multivariate Regression.

UNIT IV

Dimensionality Reduction: Introduction-Subset Selection - Principal Component Analysis - Feature Embedding - Factor Analysis - Singular Value Decomposition and Matrix Factorization-Multidimensional Scaling - Linear Discriminant Analysis - Canonical Correlation Analysis – Isomap - Locally Linear Embedding - Laplacian Eigenmaps-tDistributed Stochastic Neighbor Embedding

UNIT V

Clustering: Introduction-Mixture Densities-k-Means Clustering-Exception-Maximization Algorithm-Mixtures of Latent Variable Models-Supervised Learning after ClusteringSpectral Clustering-Hierarchical Clustering-Choosing the Number of Clusters. Non Parametric Methods: Introduction- Nonparametric Density Estimation – Generalization to multivariate data – Nonparametric Regression: Smoothing Models. Introduction- Univariate Trees-Pruning- Rules extractions from Trees- Learning rules from data – Multivariate Trees.

Core Laboratory – IV
24UAM502: Machine Learning Lab

Syllabus

1. Solving Regression & Classification using Decision
2. Root Node Attribute Selection for Decision Trees using Information Gain
3. Bayesian Inference in Gene Expression Analysis
4. Pattern Recognition Application using Bayesian Inference
5. Bagging in Classification
6. Bagging, Boosting applications using Regression Trees
7. Data & Text Classification using Neural Networks
8. Using Weka tool for SVM classification for chosen domain application
9. Data & Text Clustering using K-means algorithm
10. Data & Text Clustering using Gaussian Mixture Models
11. Implement SVM for binary classification using a real-world dataset.
12. Implement the KNN algorithm to classify handwritten digits using the MNIST.
13. Build a logistic regression model to predict exam pass/fail based on study hours and attendance.

SEMESTER V

Discipline Centric Core Course – III

24UAM503: Computer Vision

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No	CO No.	Course Outcomes	Cognitive Level
Unit -I	CO1	Understand image formation models and apply basic image processing techniques.	K1-K4
Unit -II	CO2	Apply optimization and recognition techniques for vision-based tasks.	K1 – K4
Unit -III	CO3	Analyze and implement feature detection, matching, and image alignment methods.	K1 - K4
Unit -IV	CO4	Evaluate motion estimation and computational photography techniques.	K1 – K4
Unit -V	CO5	Solutions for depth estimation and image-based rendering problems.	K1 – K5

Syllabus:

UNIT I : Fundamentals of Image Formation and Processing

Introduction-Image Formation: Geometric Primitives and Transformations, Photometric image formation- The digital Camera. Image Processing: Point Operators-Linear Filtering-More neighbourhood operators- Fourier Transforms-Pyramids and wavelets-Geometric transformations.

UNIT II: Model Optimization and Visual Recognition

Model fitting and optimization: Scattered data interpolation-Variational methods and regularization- Markov random fields. Recognition: Instance recognition-Image Classification-Object Detection-Semantic Segmentation-Video understanding -Vision and language.

UNIT III: Feature Detection and Image Alignment

Feature Detection and matching: Points and patches - Edges and contours- Contour tracking-Lines and vanishing points- Segmentation. Image Alignment and Stitching: Pairwise alignment-Image stitching - Global alignment – Compositing.

UNIT IV: Motion Estimation and Computational Photography

Motion Estimation: Translational alignment - Parametric motion- Optical flow- Layered motion. Computational Photography: Photometric calibration- High dynamic range imaging - Super-resolution, denoising, and blur removal - Image matting and compositing - Texture analysis and synthesis.

UNIT V: Depth Estimation and Image-Based Rendering

Structure from Motion and SLAM: Geometric intrinsic calibration- Pose estimation- Two-frame

structure from motion - multi-frame structure from motion- Simultaneous localization and mapping (SLAM). Depth Estimation: Epipolar geometry- 2 Sparse correspondence- Global optimization - Image Based Rendering: View interpolation - Layered depth images - Light fields and Lumigraphs- Environment mattes- Video- based rendering- Neural rendering.

Textbook:

1.Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer.

Reference Books:

1.Forsyth, D. A., & Ponce, J. (2012). *Computer vision: A modern approach* (2nd ed.). Pearson. 2.Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge University Press.
3.Prince, S. J. D. (2012). *Computer vision: Models, learning, and inference*. Cambridge University Press.
4.Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Discipline Centric Core Course – IV
24UAM504: Introduction to Deep Learning

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No	CO No.	Course Outcome	Cognitive Level
Unit -I	CO1	Understand fundamentals of neural networks and activation functions	K1 – K3
Unit -II	CO2	Apply optimization techniques for training neural networks	K1 – K3
Unit -III	CO3	Implement neural networks using PyTorch	K1 – K3
Unit -IV	CO4	Design and evaluate CNN architectures for image-related tasks	K1 – K4
Unit -V	CO5	Develop RNN-based models and understand advanced architectures like attention	K1 – K4

Syllabus

UNIT I: Fundamentals of Neural Networks and Activation Functions

The Neural Network: Building Intelligent Machines - The Limits of Traditional Computer Programs - The Mechanics of Machine Learning - The Neuron - Expressing Linear Perceptrons as Neurons - Feed-Forward Neural Networks - Linear Neurons and Their Limitations - Sigmoid, Tanh, and ReLU Neurons - Softmax Output Layers

UNIT II: Training Neural Networks and Optimization Techniques

Training Feed-Forward Neural Networks: The Fast-Food Problem - Gradient Descent - The Delta Rule and Learning Rates - Gradient Descent with Sigmoidal Neurons - The Backpropagation Algorithm - Stochastic and Minibatch Gradient Descent - Test Sets, Validation Sets, and Overfitting - Preventing Overfitting in Deep Neural Networks.

UNIT III: Neural Network Implementation using PyTorch

Implementing Neural Networks in PyTorch **:** Introduction to PyTorch - Installing PyTorch - PyTorch Tensors - Tensor Init - Tensor Attributes - Tensor Operations - Gradients in PyTorch - The PyTorch nn Module - PyTorch Datasets and Data loaders - Building the MNIST Classifier in PyTorch.

Unit IV - Convolutional Neural Networks

Convolutional Neural Networks: Neurons in Human Vision – Feature Selection – Max Pooling – Architectural Description of Convolution Networks - Closing the Loop on MNIST with Convolutional Networks - Accelerating Training with Batch Normalization - Group Normalization for Memory Constrained Learning Tasks - Building a Convolutional Network for CIFAR-10.

UNIT V: Recurrent Neural Networks and Advanced Deep Learning Models

Recurrent Neural Network – Long Short-Term Memory Units - PyTorch Primitives for RNN Models - Implementing a Sentiment Analysis Model - Solving seq2seq Tasks with Recurrent

Neural Networks - Memory augmented Neural Networks: Neural Turing Machines – Attention-Based Memory Access - NTM Memory Addressing Mechanisms.

Textbook:

1. Nithin Buduma, Nikhil Buduma, Joe Papa, Fundamentals of Deep Learning, 2nd Edition, O'Reilly Media, 2022.O'Reilly Media, Inc.

Reference Books:

1. Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning, MIT Press, 2016.
2. Stevens, E., Antiga, L., & Viehmann, T., Deep Learning with PyTorch, Manning Publications, 2020.
3. Aggarwal, C. C. Neural Networks and Deep Learning*:* A textbook. *Springer*. 2023.
4. Géron, A, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2024.

Discipline Centric Core Course – V

24UAM505: Full Stack Development

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No	CO No.	Course Outcome	Cognitive Level
Unit -I	CO1	Understand web development architecture and JavaScript fundamentals	K1 – K4
Unit -II	CO2	Apply JavaScript programming concepts to build dynamic web Applications	K1 – K4
Unit -III	CO3	Develop backend applications using Node.js and Express.js	K1 – K4
Unit -IV	CO4	Design and manage databases using MongoDB	K1 – K4
Unit -V	CO5	Build full stack applications using React / Angular and MERN stack	K1 – K4

Syllabus

Unit I – Basics of Full Stack and Java Script

Understanding the Basic Web Development Framework – User – Browser – Webserver - Backend Services - Java Script: Defining Variables - JavaScript Data Types – Operators - Implementing Looping - Creating Functions - Defining Functions - Passing Variables to Functions - Returning Values from Functions - JavaScript Objects – Working with Strings - Working with Arrays - Adding Error Handling.

Unit II - NODE JS

Basics of Node JS – Installation – Working with Node packages – Using Node package manager – Creating a simple Node.js application – Using Events – Listeners –Timers – Callbacks – Handling Data I/O – Implementing HTTP services in Node.js.

UNIT III - MONGO DB

Understanding NoSQL and MongoDB – Building MongoDB Environment – User accounts – Access control – Administering databases – Managing collections – Connecting to MongoDB from Node.js – simple applications

UNIT IV - EXPRESS AND ANGULAR

Implementing Express in Node.js – Configuring routes – Using Request and Response objects – Angular – Typescript – Angular Components – Expressions – Data binding – Built-in directives

UNIT V - REACT

MERN STACK – Basic React applications – React Components – React State – Express REST. APIs – Modularization and Webpack – Routing with React Router – Server-side rendering.

Textbooks:

1. Brad Dayley, Brendan Dayley, Caleb Dayley, Node.js, MongoDB and Angular Web Development, Addison-Wesley, Second Edition, 2018. (Unit I, II, III & IV)
2. Vasan Subramanian, 'Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node', Second Edition, Apress, 2019. (Unit – V)

Reference Books:

1. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, Apress; 1st edition, 2018.

2. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Addison-Wesley Professional, 2nd edition, 2018.

Discipline Centric Core Course - VI

24UAM506: Multi Agent System

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No	CO No.	Course Outcome	Cognitive Level
Unit -I	CO1	Understand the concepts of intelligent agents and their Architectures	K1 – K4
Unit -II	CO2	Analyze different agent types and reasoning mechanisms	K1 – K4
Unit -III	CO3	Apply agent communication and ontology concepts in distributed systems	K1 – K4
Unit -IV	CO4	Design agent-based systems using appropriate methodologies	K1 – K4
Unit -V	CO5	Evaluate multi-agent interactions using game-theoretic approaches	K1 – K4

Syllabus:

UNIT I: Foundations of Intelligent Agents and Agent Architectures

Introduction: Agents as a paradigm for software engineering- Agents as a tool for understanding human societies. Intelligent Agents: Intelligent Agents-Agents and Objects-Agents and Expert Systems-Agents as Intentional Systems-Abstract Architectures for Intelligent Agents. Deductive Reasoning Agents: Agents as Theorem Provers-Agent-Oriented Programming.

UNIT II: Practical, Reactive, and Hybrid Agent Systems

Practical Reasoning Agents: Practical Reasoning -Deliberation and Means-Ends Reasoning-Means-Ends Reasoning- Implementing a Practical Reasoning Agent-The Procedural Reasoning System. Reactive and Hybrid Agents**:** Reactive Agents-The Subsumption Architecture-The Agent Network Architecture-The Limitations of Reactive Agents- Hybrid Agents.

UNIT III: Ontology, Communication, and Coordination

Understanding Each Other: Ontology Fundamentals-Ontology Building Blocks- Ontology Languages- XML - Ad Hoc Ontologies- OWL - The Web Ontology Language-RDF-Constructing an Ontology. Communicating: Speech Acts- Searle-Agent Communication Languages-KQML-The FIPA Agent Communication Language- JADE. Working Together: Cooperative Distributed Problem Solving-Task Sharing and Result Sharing.

UNIT IV: Agent-Oriented Design Methodologies and Applications

Methodologies**:** Agent-Oriented Analysis and Design-The AAII methodology- Gaia-Tropos-Prometheus- Agent UML-Agents in Z- Pitfalls of Agent Development- Mobile Agents Applications**:** Agents for Workflow and Business Process Management-Agents for Distributed Sensing-Agents for Information Retrieval and Management- Agents for Electronic Commerce.

UNIT V: Game Theory and Decision Making

Multiagent Interactions: Utilities and Preferences-Setting the Scene-Solution Concepts and

Solution Properties- Dominant Strategies-Nash Equilibria-Pareto Efficiency-Maximising Social Welfare- Competitive and Zero-Sum Interactions-The Prisoner's Dilemma-The shadow of the future-Program Equilibria-Other Symmetric 2 x2 Interactions-Representing Multiagent Scenarios-Dependence Relations in Multiagent Systems. Making Group Decisions**:** Social Welfare Functions and Social Choice Functions.

Textbook:

1. Wooldridge, M. (2009). *An introduction to multiagent systems* (2nd ed.). John Wiley & Sons.

Reference Books:

1. Weiss, G. (Ed.). (2013). *Multiagent systems* (2nd ed.). MIT Press.
2. Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.
3. Huhns, M. N., & Singh, M. P. (1998). *Readings in agents*. Morgan Kaufmann.
4. Bordini, R. H., Hübner, J. F., & Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*. Wiley.

Discipline Centric Core Laboratory V

24UAM553: Full Stack Development Lab

Credits: 2 **Internal Marks: 40**

Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO No.	Course Outcomes	Cognitive Level
CO1	Identify the basic elements and structure used in creating interactive web systems.	K1-K3
CO2	Describe the process involved in designing user interfaces and managing interactions.	K1 - K3
CO3	Demonstrate the methods used for handling data and updating content dynamically.	K1 – K3
CO4	Explain the interaction between data management, processing logic, and presentation layers.	K1 – K4
CO5	Illustrate techniques for organizing components to build responsive and efficient web solutions.	K1 – K4

Syllabus:

JavaScript (Basics)

1. Write a program to create a simple mark sheet and calculate total and average using JavaScript.
2. Write a program to check whether a number is even or odd using JavaScript.
3. Write a program to store and display student details using JavaScript objects.
4. Write a program to perform basic string operations (length and reverse).

Node.js

5. Write a program to read and write data into a file using Node.js.
6. Write a program to create a simple HTTP server in Node.js.
7. Write a program to display a message using Node.js events.

MongoDB

8. Write a program to create a database and insert records in MongoDB.
9. Write a program to retrieve records from MongoDB.
10. Write a program to update and delete records in MongoDB.

Express & Angular

11. Write a program to create a basic Express server.
12. Write a program to handle GET request using Express.
13. Write a program to create a simple Angular component to display data.

React

14. Write a program to create a simple React application to display data.
15. Write a program to create a counter application using React state.

Tools Required:

- Editors & Browsers: Visual Studio Code, Google Chrome / Mozilla Firefox
- Front-End: JavaScript (ES6), Angular, ReactJS
- Back-End: Node.js, Express.js
- Database: MongoDB

- Package Management: npm
- Testing & Utilities: Postman, MongoDB Compass
- Version Control: Git, GitHub / GitLab

Discipline Centric Elective – I

Track 1: Higher Studies

24UAM701: Distributed Machine Learning Patterns

Credits: 4 **Internal Marks: 25**

Contact Hours: 4 **External Marks: 75**

Course Outcomes:

Unit No	CO No.	Course Outcome	Cognitive Level
Unit -I	**CO1**	Understand distributed machine learning concepts and data ingestion patterns	K1-K3
Unit -II	**CO2**	Apply distributed training, model serving, and workflow patterns	K1-K3
Unit -III	**CO3**	Analyze system architecture and implement distributed ML pipelines	K1- K4
Unit -IV	**CO4**	Apply distributed training strategies using TensorFlow and PyTorch	K1-K3
Unit -V	**CO5**	Evaluate performance and scalability of distributed machine learning systems	K1-K3

Syllabus:

UNIT I – Fundamentals of Distributed Machine Learning

Introduction to distributed machine learning systems: Large-scale machine learning-Distributed Systems- Distributed machine learning Systems.Data Ingestion Patterns: Data Ingestion-The Fashion-MNIST dataset-Batching Pattern-Sharding Pattern-Caching Pattern.

UNIT II – Distributed Machine Learning Patterns

Distributed Training Patterns: Distributed Training-Parameter Server Pattern-Collective Communication Pattern-Elastixity and fault-tolerance pattern. Model Serving Patterns: Model Serving-replicated Services Pattern-Sharded Services Pattern-The event-driven Processing Pattern. Workflow Patterns: Workflow-Fan- in and fan-out Patterns-Synchronous and Asynchronous Patterns-Step Memoization Pattern.

UNIT III – Operations and Implementation

Operation Patterns: Operations in machine learning Systems-Scheduling Patterns-Metadata Pattern.Project Overview and System Architecture: Project Overview-Data Ingestion-Model Training-Model Serving-End- to-end Workflow. Overview of Relevant Technologies: TensorFlow-Kubernetes-Kubeflow-Argo Workflows. A Complete Implementation: -Data Ingestion-Model Training-Model Serving-The end-to-end workflow.

UNIT IV – TensorFlow Distributed Training Strategies

TensorFlow Distributed Machine Learning Approach: A Quick Overview of TensorFlow-TensorFlow Cluster Process Roles and Responsibilities- Loading Parquet Data into a TensorFlow Dataset- An Inside Look at TensorFlow's Distributed Machine Learning Strategies-Parameter Server Strategy- Central Storage Strategy- One Machine, Multiple Processors-Mirrored Strategy-One Machine, Multiple Processors, Local Copy Multi Worker Mirrored Strategy: Multiple

Machines, Synchronous.

UNIT V – PyTorch Distributed Training Approaches

PyTorch Distributed Machine Learning Approach: A Quick Overview of PyTorch Basics- Computation Graph- PyTorch Mechanics and Concepts- PyTorch Distributed Strategies for Training Models- Introduction to PyTorch's Distributed Approach- Distributed Data-Parallel Training- RPC-Based Distributed Training- Communication Topologies in PyTorch.

Textbooks:

1. Tang, Y. (2024). Distributed machine learning patterns, First Edition. Manning Publications. (ISBN: 9781617299025).
2. Polak, A. (2023). Scaling machine learning with spark: Distributed ML with MLlib, TensorFlow, and PyTorch (1st ed.). O'Reilly Media.

Reference Books:

1. Kane, F. (2020). Hands-on machine learning engineering with Python: Design and build scalable machine learning systems. Packt Publishing.
2. Burkov, A. (2020). Machine learning engineering. True Positive Inc.
3. Grant, T., & Bradshaw, J. (2019). Kubeflow for machine learning. Packt Publishing.
4. Ramsundar, B., & Zadeh, R. B. (2018). TensorFlowfor deep learning. O'Reilly Media.

Discipline Centric Elective Laboratory- I

24UAM751: Open AI Application Programming Interface Lab

Credits: 2 **Internal Marks: 40**

Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO No.	Course Outcome	Cognitive Level
CO1	Understand the usage of AI APIs for different applications	K3
CO2	Develop text-based applications using GPT API	K3
CO3	Implement audio processing applications using speech APIs	K3
CO4	Create and manipulate images using DALL·E API	K4
CO5	Build integrated AI applications combining text, audio, and image processing	K4

List of Exercises:

Using GPT API (Text Based Programs)

1. Program to generate text for a given prompt using GPT API.
2. Program to generate a paragraph based on user input using GPT API.
3. Program to develop a simple chatbot using GPT API.
4. Program to summarize a paragraph using GPT API.
5. Program to perform grammar correction using GPT API.
6. Program to generate code (e.g., sorting, searching) using GPT API.
7. Program to perform sentiment analysis using GPT API.
8. Program to develop a Question Answering system using GPT API.
9. Program to generate formal email using GPT API.
10. Program to translate text from English to Tamil using GPT API.

Using OpenAI Audio API

11. Program to convert speech to text using OpenAI Audio API.
12. Program to convert text to speech using OpenAI Audio API.

Using DALL-E API (Image Generation)

13. Program to generate an image from text description using DALL-E API.
14. Program to generate different variations of an image using DALL-E API.

15. Program to edit an image using a text prompt using DALL-E API.

Tools Required:

- AI Platforms / APIs: OpenAI API
- Programming Languages: Python / JavaScript
- Development Environment: Visual Studio Code
- API Testing Tools: Postman
- Libraries & Frameworks: OpenAI SDK (Python/Node.js), Requests (Python)

Track 2: Employment

24UAM703: MLOps & AI Deployment

Credits: 4 **Internal Marks: 25**
Contact Hours: 4 **External Marks: 75**

Unit No	CO No.	Course Outcome	Cognitive Level
Unit -I	CO1	Knowledge of fundamental concepts, procedures, and tools used okin data analytics and programming in R.	K1-K4
Unit -II	CO2	Familiarity with techniques for preparing, cleaning, and managing datasets effectively.	K1 – K4
Unit -III	CO3	Ability to present data visually for clear communication of patterns and insights.	K1 – K4
Unit -IV	CO4	Competence in performing statistical and predictive modelling using R methods.	K1 – K5
Unit -V	CO5	Awareness of producing structured reports based on results and data insights.	K1 – K4

Syllabus

UNIT I: MLOps Fundamentals

Foundations of MLOps MLOps Evolution - DevOps vs. MLOps - ML Production Challenges - ML Lifecycle Management - Key Tools and Frameworks Overview - Cloud Computing Tools: AWS ML Tools, Google Cloud AI, IBM Watson - Representation Learning: The Core of Modern AI.

UNIT II: Model Development

Versioning, Pipelines, and Preprocessing Data Versioning with DVC - Model Versioning using MLflow - Pipeline Orchestration using Kubeflow - Automated Data Preprocessing and Validation - Experiment Tracking - Data Preprocessing for Language Models: Text Vectorization Layer - Standardization, Vocabulary Indexing - Embedding Word Vectors - TF-IDF - Tokenization Fundamentals: Byte Pair Encoding - Bag of Words Model and Sequential Models.

UNIT III: Deployment & Monitoring

Strategies and Transformers Deployment Strategies: Batch, Online, Streaming - Serving Models with FastAPI and Flask - Dockerizing ML Models - Kubernetes Basics for ML Deployment - Serverless Deployment - AWS Lambda, Google Cloud Functions - Transformer Encoder for Language Comprehension - BERT Models - Decoder-only GPT Models for Language Generation - Generative Adversarial Networks - Diffusion Models for Image Generation - Autoencoders for Representation Learning - PreTraining - Reinforcement Learning through Human Feedback.

UNIT IV: Governance & Responsible AI

CI/CD and LLMOps Continuous Integration and Continuous Delivery/Deployment (CI/CD) for ML Systems - Feature Stores - Testing ML Pipelines -Unit, Integration, Regression - Responsible

and Ethical MLOps - Real-World Case Studies -Retail, Finance, Healthcare - LLMOps: Tools and Platforms - LangChain, OpenAI API - Prompt Engineering, Retrieval-Augmented Generation (RAG) - Low-Rank Adaptation (LoRA) and its variants - Guardrails and Processing of LLM Outputs.

UNIT V: MLOps Applications

Monitoring and Governance Model Monitoring: Accuracy, Drift, Bias - Logging and Observability Tools - Data Drift Detection - Model Retraining Pipelines - Governance and Compliance in ML

Textbook:

Treveil, M., & Dataiku Team. (2021). *Mastering MLOps: How to scale machine learning in the enterprise.* O'Reilly Media.

Reference Books:

1. Gift, N., & Deza, A. (2021). *Practical MLOps: Operationalizing machine learning models.* O'Reilly Media.
2. Lakshmanan, V., Robinson, S., & Munn, M. (2020). *Machine learning design patterns: Solutions to common challenges in data preparation, model building, and MLOps*. O'Reilly Media.
3. Ameisen, E. (2020). *Building machine learning powered applications: Going from idea to product.* O'Reilly Media.

24UAM752: MLOps Tools Lab

Credits: 2 **Internal Marks: 40**

Contact Hours: 4 **External Marks: 60**

Course Outcomes:

CO No.	Course Outcomes	Cognitive Level
CO1	Understand basic ML workflow and data preprocessing using Scikit-learn and Pandas	K1- K3
CO2	Apply EDA, feature scaling, and build ML models	K1- K3
CO3	Evaluate and compare ML models using performance metrics	K1- K4
CO4	Apply MLOps practices like versioning and experiment tracking using MLflow	K1- K4
CO5	Deploy and monitor ML models using Flask and explain results using SHAP	K1-K4

List of Exercises:

BASIC ML WORKFLOW (FOUNDATION)

1. Write a program to implement a basic Machine Learning workflow using Scikit-learn to load dataset, train and test a model.
2. Write a program to split a dataset into training and testing sets using Scikit-learn.
3. Write a program to implement data preprocessing (handling missing values and normalization) using Pandas.

MODEL DEVELOPMENT

4. Write a program to perform Exploratory Data Analysis (EDA) using Matplotlib to visualize dataset features.
5. Write a program to implement feature scaling using StandardScaler in Scikit-learn.
6. Write a program to train a classification model using Logistic Regression.
7. Write a program to train and compare two models (e.g., Decision Tree and KNN) using Scikit-learn.
8. Write a program to evaluate a model using accuracy, precision, and recall metrics.

MLOps PRACTICES

9. Write a program to save and load a trained model using pickle (model versioning).
10. Write a program to track machine learning experiments using MLflow (basic logging).
11. Write a program to ensure reproducibility by setting random seed and running model multiple times.

DEPLOYMENT

12. Write a program to deploy a machine learning model using Flask.
13. Write a program to create a simple API that takes input and returns prediction.

MONITORING & RESPONSIBLE AI

14. Write a program to detect data drift by comparing mean values of training and new data.
15. Write a program to explain model predictions using SHAP.

Tools Required:

- Programming: Python 3.x, Jupyter Notebook / Google Colab
- Libraries: Scikit-learn, Pandas, Matplotlib
- MLOps Tools: MLflow, Pickle, Flask, SHAP
- Supporting Tools: Git & GitHub, Web Browser / Postman

Ability Enhancement Course

Technical Skill

24UAM381: Prompt Engineering

Credits: 2 **Internal Marks: 40**
Contact Hours: 2 **External Marks: 60**

Course Outcomes:

CO No.	Course Outcomes	Cognitive Level
CO1	Understand and construct basic prompts for academic tasks	K1, K3
CO2	Apply prompt engineering techniques for programming and debugging	K1 - K3
CO3	Generate structured outputs for reports, presentations, and case studies	K1 - K4
CO4	Utilize AI tools for career preparation and professional development	K1 – K4
CO5	Create innovative solutions using AI for creative and practical applications	K1 – K4

Assignments & Presentation

1. Write a prompt to generate a seminar presentation (PPT content) using Google Gemini.
2. Write a prompt to create a well-structured assignment/report.
3. Write a prompt to generate case study with real-life examples.
4. Write a prompt to convert content into bullet points and diagrams/flow.

Programming & Technical Skills

1. Write a prompt to generate a Python program with explanation using Microsoft Copilot.
2. Write a prompt to debug and correct errors in code.
3. Write a prompt to generate SQL queries for a database problem.
4. Write a prompt to explain program logic step-by-step.

Career & Placement Preparation

1. Write a prompt to create a resume and cover letter.
2. Write a prompt to generate technical interview questions and answers.
3. Write a prompt to conduct a mock interview (Q&A format).
4. Write a prompt to suggest mini project ideas with tools and outcomes.

Creative & Practical Skills

1. Write a prompt to generate a poster or logo idea using Midjourney.
2. Write a prompt to generate social media content/post.
3. Write a prompt to create a simple business idea or startup plan.
4. Write a prompt to design a basic chatbot conversation flow using LangChain.

Tools Required

- AI Tools / Platforms: Google Gemini, Microsoft Copilot, ChatGPT
- Programming & Database: Python, SQL
- Development Environment: Visual Studio Code
- Creative Tools: Midjourney, LangChain
- Documentation & Presentation: Microsoft Word / Google Docs, Microsoft PowerPoint / Google Slides
- Web Browser: Google Chrome / Mozilla Firefox