B. V. V. Sangha's
Basaveshwar Engineering College, Bagalkote
Department of Artificial Intelligence and Machine Learning

3rd Semester Scheme and Syllabus

Sl. No.	Course Category	Course Code	Course Title	Teaching-Learning Scheme				Examination Duration (Hrs/Sem.)		Total Hours	Examination Scheme and Evaluation			Credits (C)=Total Hours/30
				L Hrs/Week (Hrs/Sem.)	T Hrs/Week (Hrs/Sem.)	P Hrs/Week (Hrs/Sem.)	Term Work (TW) & Self	CIE	SEE		CIE Marks	SEE Marks	Total Marks	
1.	BSC	1BMAS301C	Mathematics – III for Computer Science	3 (42)	0	0	42	3	3	90	50	50	100	3
2.	PCC	1BAIA302C	Data Structures and Applications	3 (42)	0	0	42	3	3	90	50	50	100	3
3.	PCC	1BAIA303C	Digital Design and Computer Organization	2 (28)	0	0	26	3	3	60	50	50	100	2
4.	PCC	1BAIA304C	Principles of AI	3 (42)	0	0	42	3	3	90	50	50	100	3
5.	IPCC	1BAIA305C	Fundamentals of Data Science	2(28)	0	2(26)	28	5	3	90	50	50	100	3
6.	PCC	1BAIA306C	Embedded Systems	2(28)	0	0	26	3	3	60	50	50	100	2
7.	PCC	1BAIA306L	Data Structures Laboratory	0	0	2 (26)	0	2	2	30	50	50	100	1
8.	IPCC	1BAIA307C	OOPs with JAVA	2(28)	0	2(26)	28	5	3	90	50	50	100	3
9.	MC	1BHSB36M /1BHSC36M / 1BHSA36M	NSS-II / P.Ed-II / Yoga-II	1	0	0	14	2	–	30	25	–	25	0
			Total	18		6				600	425	400	825	20

<table>
<tr><td colspan="2">Course Code</td><td>:</td><td>1BMAS301C</td><td rowspan="6">Mathematics-III for Computer Science Stream</td><td>Semester</td><td>:</td><td>3</td></tr>
<tr><td colspan="2">L:T:P</td><td>:</td><td>3:0:0</td><td>Course Type</td><td>:</td><td>BSC</td></tr>
<tr><td rowspan="4">Hours/Sem.</td><td>Teaching</td><td>:</td><td>42 Hrs</td><td>CIE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Learning (TW+SL)</td><td>:</td><td>42 Hrs</td><td>SEE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Exam</td><td>:</td><td>06 Hrs</td><td>Total Marks</td><td>:</td><td>100</td></tr>
<tr><td>Total Hrs.</td><td>:</td><td>90 Hrs</td><td>Credits</td><td>:</td><td>03</td></tr>
</table>

Professional Competency:

Apply probability and statistical techniques to solve complex engineering problems

Course Outcomes

At the end of the course, the student will be able to

CO1: Apply statistical data for correlation and regression techniques to develop suitable mathematical models

CO2: Apply the concepts of probability and probability distribution.

CO3: Apply the notion of a discrete-time Markov chain and n-step transition probabilities to solve the given problem

CO4: Apply the Statistical Inference related to engineering problems.

UNIT-I	10 Hrs.

Curve fitting, Correlation and Regressions: Principles of least squares, Curve fitting by the method of least squares in the form $y = a + bx$, $y= a+ bx + cx^2$, $y = ab^x$. Correlation, Regression and Rank Correlation.

UNIT–II	10 Hrs.

Probability Distributions: Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation, mean and variance. Binomial, Poisson and Normal distributions- problems (derivations for mean and standard deviation for Binomial and Poisson distributions only)-Illustrative examples.

UNIT–III	10 Hrs.

Joint Probability distribution: Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. **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.

UNIT–IV	10 Hrs.

Statistical Inference : Introduction, sampling distribution, standard error, testing of hypothesis, levels of significance, test of significances, confidence limits, simple sampling of attributes, test of significance for large samples, comparison of large samples. Test of Significance for means of two small samples, students‘t’ distribution, Chi-square distribution as a test of goodness of fit. F-Distribution.

Text Books:

1. **Ronald E. Walpole, Raymond H Myers, Sharon L Myers & Keying Ye** “Probability & Statistics for Engineers & Scientists”, Pearson Education, 9th edition, 2017.
2. **Peter Bruce, Andrew Bruce & Peter Gedeck** “Practical Statistics for Data Scientists” O’

Reilly Media, Inc., 2nd edition **2020**.
Reference Books:

1. Erwin Kreyszig, "Advanced Engineering Mathematics", John Wiley & Sons, 9th Edition , 2006.
2. B. S. Grewal "Higher Engineering Mathematics", Khanna publishers, 44th Edition, 2021.
3. G Haribaskaran "Probability, Queuing Theory and Reliability Engineering", Laxmi Publication, Latest Edition, 2006.
4. Irwin Miller & Marylees Miller, JohnE. Freund's "Mathematical Statistics with Applications "Pearson. Dorling Kindersley Pvt.Ltd.India, 8th Edition, 2014.
5. S C Gupta and V K Kapoor, "Fundamentals of Mathematical Statistics", SChandand Company, Latest Edition.
6. Robert V. Hogg, Joseph W. McKean & Allen T. Craig, "Introduction to Mathematical Statistics", Pearson Education 7th Edition, 2013.
7. Jim Pitman, "Probability", Springer-Verlag, 1993.
8. Sheldon M. Ross, "Introduction to Probability Models" 11th Edition, Elsevier, 2014.
9. A. M. Yaglom and I. M. Yaglom, "Probability and Information" .D. Reidel Publishing Company, Distributed by Hindustan Publishing Corporation (India) Delhi,1983.
10. P. G. Hoel, S.C. PortandC. J. Stone, "Introduction to Probability Theory", Universal Book Stall, (Reprint), 2003.
11. S.Ross, "A First Course in Probability", Pearson Education India, 6th Ed.,2002.
12. W. Feller, "An Introduction to Probability Theory and its Applications", Vol. 1,Wiley, 3rd Ed., 1968.
13. N. P. Baliand Manish Goyal, A Text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
14. Veerarajan T, Engineering Mathematics (for semester III), Tata McGraw-Hill, New Delhi.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Program Outcomes and (WKs)											Program Specific Outcomes (PSOs)	
	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3 (WK1) (WK2) (WK3) (WK4)	2 (WK1) (WK2) (WK3)						1			1		
CO2	3 (WK1) (WK2) (WK3) (WK4)	2 (WK1) (WK2) (WK3)						1			1		
CO3	3 (WK1) (WK2) (WK3) (WK4)	2 (WK1) (WK2) (WK3)						1			1		
CO4	3 (WK1) (WK2)	2 (WK1) (WK2)						1			1		

	(WK3) (WK4)	(WK3											

Course Code		:	1BAIA302C	Data Structures and Applications	Semester	:	3
L:T:P		:	3:0:0		Course Type	:	PCC
Hours/Sem.	Teaching	:	42 Hrs		CIE Marks	:	50
	Learning (TW+SL)	:	42 Hrs		SEE Marks	:	50
	Exam	:	06 Hrs		Total Marks	:	100
	Total Hrs.	:	90 Hrs		Credits	:	03

Professional Competency:	
Data structures helps in writing efficient programs, choose the right structure for the Solving complex computational problems and Building scalable software systems	
Course Outcomes:	
After completion of the course, student will be able to: **CO1;** Understand the fundamentals of data structures and perform operations on arrays, structures, unions, and sparse matrices. **CO2:** Apply stack and queue concepts for solving problems such as expression evaluation, conversion, and memory-efficient implementations. **CO3:** Implement and analyse linked list operations including singly linked lists, doubly linked lists, and their applications. **CO4:** Demonstrate tree concepts including binary trees, traversals, binary search trees, threaded trees, and disjoint set representations. **CO5:** Analyse graph operations, hashing techniques, priority queues, and efficient binary search tree concepts for problem-solving.	
UNIT–I	**10 Hrs.**
Introduction To Data Structures: Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations **Review of** pointers and dynamic Memory Allocation, **Arrays And Structures:** Arrays, Dynamic Allocated Arrays, Structures and Unions, Sparse Matrices, representation of Multidimensional Arrays, Strings **Stacks:** Stacks, Stacks Using Dynamic Arrays, Evaluation and conversion of Expressions	
UNIT–II	**11 Hrs.**
Queues: Queues, Circular Queues, Using Dynamic Arrays, Multiple Stacks and queues. **Linked Lists :** Singly Linked, Lists and Chains, Representing Chains in C, Linked Stacks and Queues, Polynomials , Additional List Operations, Sparse Matrices, Doubly Linked List.	
UNIT-III	**11 Hrs.**
Trees: Introduction, Binary Trees, Binary Tree Traversals, Threaded Binary Trees., Binary Search trees, Selection Trees, Forests, Counting Binary Trees, Representation of Disjoint sets	
UNIT-IV	**10 Hrs.**
Graphs: The Graph Abstract Data Types, Elementary Graph Operations **Hashing:** Introduction, Static Hashing, Dynamic Hashing **Priority Queues:** Single and double ended Priority Queues, Leftist Trees **Introduction To Efficient Binary Search Trees:** Optimal	

Binary Search Trees
Text Books:
1. llis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014
Reference Books:
1. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014. 2. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014. 3. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012. 4. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Ed, McGraw Hill, 2013 5. A M Tenenbaum, Data Structures using C, PHI, 1989Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	**Programme Outcomes (POs) and WKs**											**Program Specific Outcomes (PSOs)**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	**2 WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1 (WK8)**	**2**	**1**	**-**
CO2	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**-**	**3 (WK2) (\WK6)**	**1 (WK1) (WK5) (WK7)**	**1 (WK9)**	**1**	**1**	**-**	**1**	**2**	**2**	**-**
CO3	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) WK2) (WK3) (WK4)**	**2 (WK5)**	**-**	**-**	**1 (WK1) (WK5) (WK7)**	**-**	**1**	**1**	**-**	**1**	**2**	**2**	**-**
CO4	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**-**	**-**	**1 (WK1) (WK5) (WK7)**	**-**	**1**	**1**	**-**	**1**	**2**	**2**	**1**
CO5	**-**	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK5)**	**-**	**-**	**1 (WK1) (WK5) (WK7)**	**-**	**1**	**1**	**-**	**1 (WK8)**	**-**		**2**

Course Code		:	1BAIA303C	**Digital Design and Computer Organization**	Semester	:	3
L:T:P		:	2:0:0		Course Type	:	PCC
Hours/Sem.	Teaching	:	28 Hrs		CIE Marks	:	50
	Learning (TW+SL)	:	26 Hrs		SEE Marks	:	50
	Exam	:	06 Hrs		Total Marks	:	100
	Total Hrs.	:	60 Hrs		Credits	:	02

professional Competency:

Apply digital design and computer organization concepts to develop and analyze combinational and sequential circuits.Use hardware description tools and processor architecture principles to solve computing system design problems.

Course Outcomes:

After completion of the course, student will be able to:
CO1:Design combinational circuits using Boolean algebra, K-maps, and HDL.
CO2:Develop sequential circuits with latches, flip-flops, counters, and registers.
CO3:Explain instructions, addressing modes, I/O and memory hierarchy.
CO4:Perform arithmetic operations and analyze instruction.

UNIT–I	07 Hrs.

Digital Design Fundamentals: Binary Logic, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates, **Karnaugh Map Method (K-map)**: Three-Variable Map, Four-Variable Map, Don't-Care Conditions. Quine- Mccluskey Method. **Hardware Description Language (HDL)**: Verilog Model of a Simple Circuits, **Introduction to Combinational Circuits**: Design Procedure, Binary Adder-Subtractor, Decoders, Encoders, Multiplexers.

UNIT–II	07 Hrs.

Sequential Logic: Introduction to Sequential Circuits, Basic Architectural Distinctions between Combinational and Sequential circuits, **Storage Elements**: Latches, **Flip-Flops:** SR, JK, D, T and master slave, characteristic tables and equations. **Counters:** Design of Single mode Counter, Ripple Counter, Ring Counter, Shift Register, Ring Counter Using Shift Register. **Basic Structure of Computer:** Functional Units, Basic Operational Concepts.

UNIT-III	07 Hrs.

Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instruction and Instruction Sequencing, Addressing Modes. **Input/Output Organization:** Accessing I/O Devices, **Interrupts**: Interrupt Hardware, Enabling/Disabling Interrupts, Handling Multiple Devices. **Direct Memory Access (DMA)**: Bus Arbitration.

UNIT-IV	07 Hrs.

Arithmetic Unit: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers. **Basic Processing Unit**: Register Transfers, Performing ALU Operations, Fetching a Word from Memory, Storing a Word in Memory, Execution of a Complete

Instruction.
Text Books:
1. M.Morris Mano & Michae lD. Ciletti, “Digital Design with an Introduction to Verilog Design”, 5th Edition, Pearson Education. 2.Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, 5th Edition, TataMcGraw Hill. 3. Donald D Givone, “Digital Principles and Design”, McGraw Hill Edition 2002.
Reference Books:
1. J. P. Hayes,1998, “Computer Architecture and Organization”, 3th Edition, MGH. **2.** William Stallings, 2007, “Computer Organization and Architecture”, 7th Edition, PHI.

Sl. No	Lab Assignments
1	Given a 4-variable logic expression, simplify it using appropriate technique and simulate the same using basic gates.
2	Design a 4 bit full adder and subtractor and simulate the same using basic gates.
3	Design Verilog HDL to implements circuits using structural, Data flow and Behavioural model.
4	Design Verilog HDL to implement Binary Adder – Subtractor Half and Full Adder, Half and Full Subtractor.
5	Design Verilog HDL to implement 2:4 and 3:8 decoder.
6	Design Verilog program to implement Different types of multiplexer like 2:1, 4:1 and 8:1.
7	Design Verilog program to implement types of De-Multiplexer.
8	Design Verilog program for implementing various types of Flip-Flops such as SR, JK and D.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
1.	3 (WK2) (WK3) (WK4)	2 (WK2) (WK3) (WK4)	3 (WK3) (WK4) (WK5)	1 (WK2) (WK4) (WK8)	3 (WK2) (WK6)	-	-	-	-	-	2 (WK6)	3	2	1
2.	3 (WK3) (WK4)	3 (WK2) (WK3) (WK4)	3 (WK3) (WK4) (WK5)	2 (WK2) (WK4) (WK8)	3 (WK6)	-	-	2	-	-	2 (WK6)	3	2	2
3.	3 (WK1) (WK3) (WK4)	2 (WK2) (WK3) (WK4)	2 (WK3) (WK4)	2 (WK2) (WK4) (WK8)	2 (WK6)	2 (WK1) (WK5) (WK7)	-	-	2	2 (WK9)	2 (WK6)	3	2	2
4.	3 (WK2)	3 (WK2)	2 (WK3)	2 (WK2)	2 (WK6)	-		2	2	2 (WK9)	3 (WK6)	3	2	3

Course Code		:	1BAIA304C	Principles of AI	Semester	:	3
L:T:P		:	3:0:0		Course Type	:	PCC
Hours/Sem.	Teaching	:	42 Hrs		CIE Marks	:	50
	Learning (TW+SL)	:	42 Hrs		SEE Marks	:	50
	Exam	:	06 Hrs		Total Marks	:	100
	Total Hrs.	:	90 Hrs		Credits	:	03

	(WK3) (WK4)	(WK3) (WK4)	(WK4) (WK5)	(WK4) (WK8)										

Professional Competency:

Develop the ability to analyze, design, and implement Artificial Intelligence techniques for intelligent problem solving using search algorithms, knowledge representation, logical reasoning, and uncertainty handling to build intelligent systems for real-world applications.

Course Outcomes:

After completion of the course, student will be able to:

CO1: Understand the fundamental concepts of Artificial Intelligence, intelligent agents, rational behavior, and problem formulation in diverse environments.

CO2: Analyze and apply uninformed and informed search algorithms, including heuristic-based techniques, to solve AI search and optimization problems efficiently.

CO3: Model and represent knowledge using propositional and first-order logic, and apply logical reasoning techniques to develop knowledge-based intelligent systems.

CO4: Evaluate and implement inference mechanisms in first-order logic, including unification, forward chaining, backward chaining, and resolution, and apply uncertainty reasoning techniques for intelligent decision-making under uncertain environments.

UNIT–I	10 Hrs.

Introduction: What is AI? Foundations and History of AI. **Intelligent Agents:** Agents and environment, Concept of Rationality, The nature of environment, The structure of agents.

UNIT–II	11 Hrs.

Problem solving: Problem solving agents, Example problems, Searching for Solutions, Uninformed Search. **Strategies:** Breadth First search, Depth First Search, Iterative deepening depth first search.

UNIT-III	11 Hrs.

Informed Search Strategies: Heuristic functions, Greedy best first search, A*search. Heuristic Functions. **Logical Agents:** Knowledge–based agents, The Wumpus world, Logic, Propositional logic, Reasoning patterns in Propositional Logic.

UNIT-IV	10 Hrs.

First Order Logic: Representation Revisited, Syntax and Semantics of First Order logic, Using First Order logic. **Inference in First Order Logic:** Propositional Versus First Order Inference, Unification, Forward Chaining, Backward Chaining, Resolution. **Introduction to Uncertain Knowledge and Reasoning:** Quantifying Uncertainty: Acting under Uncertainty.

Text Books:
1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence", 3rd Edition, Pearson, 2015.
Reference Books:
1. Elaine Rich, Kevin Knight, "Artificial Intelligence", 3rd Edition, Tata McGraw Hill, 2013.
2. George F Lugar, "Artificial Intelligence Structure and strategies for complex", Pearson Education, 5th Edition, 2011.

Table 5: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3 (WK1) (WK3) (WK4)	2 (WK1) (WK3) (WK4)	-	-	1 (WK6)	-	-	-	1	-	2 (WK8)	3	2	2
CO2	3 (WK2) (WK3) (WK4)	3 (WK2) (WK3) (WK4)	2 (WK5)	1 (WK8)	2 (WK2) (WK6)	-	-	-	1	-	2 (WK8)	3	3	2
CO3	3 (WK2) (WK4) (WK6)	3 (WK2) (WK4) (WK6)	2 (WK5)	2 (WK8)	2 (WK2) (WK6)	-	-	-	1	-	2 (WK8)	3	3	2
CO4	3 (WK1) (WK2) (WK4)	3 (WK1) (WK2) (WK4)	2 (WK5)	2 (WK8)	1 (WK2) (WK6)	1 (WK1) (WK5) (WK7)	1 (WK9)	-	1	-	3 (WK8)	3	3	3

Course Code		:	1BAIA305C	Fundamentals of Data Science	Semester	:	3
L:T:P		:	2:0:2		Course Type	:	IPCC
Hours/Sem.	Teaching	:	28+26=54 Hrs		CIE Marks	:	50
	Learning (TW+SL)	:	28 Hrs		SEE Marks	:	50
	Exam	:	08 Hrs		Total Marks	:	100
	Total Hrs.	:	90 Hrs		Credits	:	03

Professional Competency:

Apply data science and machine learning principles to collect, preprocess, analyze, and model real-world data, and develop, evaluate, and deploy AI/ML solutions using appropriate tools and frameworks to solve industry-relevant problems at an entry-level professional role.

Course Outcomes:

After completion of the course, students will be able to:

CO1: Analyze the significance of data in AI, compare traditional software development and AI software development life cycles, and classify machine learning types, workflows, applications, and challenges.

CO2: Apply data science and big data concepts to collect, preprocess, integrate, reduce, transform, and explore structured and unstructured datasets using appropriate tools and techniques.

CO3: Design and implement machine learning pipelines involving regression, classification, feature engineering, data splitting, model evaluation, and optimization using Python libraries.

CO4: Build, evaluate, and tune supervised and unsupervised learning models including linear regression, KNN, decision trees, logistic regression, and clustering for real-world datasets.

UNIT-I	14 Hrs.

Significance of data in AI, AI Software Development life cycle, Compare traditional software development with AI Software Development. Machining Learning, Machine learning types, Machine learning workflow, Machine learning applications, Challenges in ML, Building a model-steps involved. **Pipelines:** Data engineering, Machine learning, Deployment. Introduction to Data Science, Data Science uses, Data Science tools and technique. **Big Data**: Vs of Big Data - Sources of data, Role of Big Data in AI&ML. **Data**: Introduction, Data types: Structured Data, Unstructured Data, Challenges with Unstructured Data. **Data Preprocessing:** Importance of data preprocessing, Data cleaning , Assess Data quality , Data anomalies, Detect missing values with pandas data frame functions: info() and .isna() , Diagnose type of missing values with visual and statistical methods (eg. chi-squared test of independence). **Approaches to deal with missing values**: Keep the missing value as is, Remove data objects with missing values, Remove the attributes with missing values, Estimate and impute missing values. **Detecting outliers**: Uni-variate outlier detection, bi-variate outlier detection, Time series outlier detection. **Dealing with outliers**: Do nothing, Replace with the upper cap or lower cap, Perform a log transformation, Remove data objects with outliers. **Data transformation**: Need for data transformation, Normalization, Standardization Data transformation with - binary coding, ranking transformation and discretization

UNIT-II	14 Hrs.
Exploratory data analysis: Overview, EDA goals and benefits. **Univariate data analysis**: Characterizing data with descriptive statistics, Univariate distribution, Univariate comparison plots , Univariate composition plots. **Univariate analysis tests**: Hypothesis testing Error, Test statistic, type, interpreting test statistics. Understanding p-value. **Multivariate analysis**: Finding relationship in data using Covariance and Correlation. Multivariate distribution plot, Multivariate comparison plot, Multivariate relationship plot, Multivariate composition plot. Feature Engineering**,** Data Splitting Importance of data splitting - Training set - Validation set - Testing set, Underfitting and Overfitting **Techniques:**	
UNIT-III	**14 Hrs.**
Machine Learning pipeline, Supervised Learning: Regression, Types of regression, Regularization in ML, Real-Life Applications. **Linear regression Overview:** Types, Simple linear regression, Multiple linear regression, Polynomial linear regression, Applications of Linear Regression. Understanding Simple linear regression, Regression equation, Assumptions, Gradient descent, Setting up the regression problem. **Implementation**: Student score based on study hours Problem statement, Create a model to analyses the relation between CIE and SEE result using sklearn. Create a model to analyze the relation between crop yield and rain fall rate, Build linear regression model using Stats model. **Model Evaluation & Testing:** Evaluate regression model, Evaluation Metric, Coefficient of Determination or R-Squared (R2), Root Mean Squared Error (RSME). Optimize regression model, Gradient descent. **Cross-validation**: Why do we need Cross-Validation? Techniques - Hold out method - Leave One Out Cross-Validation - K-Fold Cross-Validation. **Multiple Linear Regression**: Overview, Assumptions, Normal Equation, Applications. Identification and collection of regression dataset, Perform data exploration, preprocessing and splitting on datasets , build regression model , evaluate the model , minimize the cost function using Boston housing price dataset from sci-kit learn datasets. Overfitting vs underfitting in Linear regression. **Supervised learning – classification, Types**: Binary classification, Multi-Label Classification, Multi-Class Classification, Imbalanced Classification, Classification models, Applications. **KNN Classification**: Overview, KNN classification and regression, Choosing best K using validation method, Perform classification on Breast cancer data set using sklearn. Evaluation Metrics for Classification - confusion matrix, Accuracy, Precision and Recall, Specificity, F1-score, AUC-ROC.	
UNIT-IV	**14 Hrs.**
Decision tree, Understanding Entropy, information gain, Issues in decision tree, Overfitting in decision tree classifier and Pruning, Decision Tree Classifier Applications. Build decision tree-based model in Python for like Play Tennis dataset from sci-kit learn Or any classification dataset from UCI, Kaggle. Evaluation of decision tree model with different metrics. Hyper parameter tuning for Decision Tree Classifier. **Logistic regression**: Introduction to logistic regression. Difference between linear and logistic regression. Applications of logistic regression. The Logistic Function. The Logistic Regression Model. Gradient Descent and Optimization. Model Evaluation. Model Validation. Implementing Logistic Regression in Python (sklearn) for real world problems. **Unsupervised Learning**: Definition and differences from supervised learning, Applications of unsupervised learning. Types of Unsupervised Learning. **Overview of Clustering**: Definition and types of clustering, Applications of clustering in different fields. **Introduction to K-Means Clustering**: Concept of K-Means Clustering. History and development of the K-Means algorithm, Real-world applications. **Understanding the K-Means Algorithm**: The objective function of K-Means, Steps involved in the K-Means algorithm.	

Distance Metrics: Euclidean distance and its importance in K-Means, Other distance metrics (Manhattan, Cosine), Choosing the right distance metric. Practical Implementation in Python (sklearn). Evaluating and Validating Clusters.
Textbooks:
1. Tom Mitchel, "Machine Learning ", International Edition 1997, McGraw Hill Education. **2.** Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow , O'REILLY publications, 3rd Edition. **3.** Wes McKinney, "Python for Data Analysis", O'REILLY publications, 2ndEdition.. **4.** Online Courses: Coursera (Andrew Ng's Machine Learning), edX, Udacity. **5.** Tutorials and Blogs: Towards Data Science, Kaggle kernels.
Reference Books:
1. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (3rd ed.). O'Reilly Media. 2. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media. 3. Ng, A. Machine Learning (Online Course). Coursera. **4.** edX. Machine Learning and Artificial Intelligence Courses. Online Learning Platform.

Data Science Lab

SL. NO.	List of Assignments
1.	**Assignment 1: Data Collection & Understanding Dataset** **Problem Statement** Collect a real-world dataset in CSV, Excel, or JSON format and perform basic data understanding using Python libraries. Identify the dataset structure, features, and target variables. Generate statistical summaries and inspect the dataset using Pandas functions. **Objectives** • Understand different types of datasets • Load datasets using Python • Explore dataset structure and attributes **Requirements** • Import dataset using Pandas • Identify structured/unstructured data • Use .head(), .info(), and .describe() • Identify feature variables and target variable **Expected Outcome** • Dataset summary report • Data type analysis • Structure of dataset
2.	**Assignment 2: Data Preprocessing – Missing Values Handling** **Problem Statement** Perform preprocessing on a dataset containing missing values. Detect null values and apply different techniques to handle incomplete data efficiently. **Objectives**

	• Detect and analyze missing data • Apply preprocessing techniques for data cleaning **Requirements** • Use .isna() and .info() to identify missing values • Remove rows/columns containing missing values • Apply mean/median imputation • Perform forward fill and backward fill techniques **Expected Outcome** • Comparison of dataset before and after preprocessing • Cleaned dataset without missing values
3.	**Assignment 3: Outlier Detection and Treatment** **Problem Statement** Identify outliers present in a dataset using statistical and visualization techniques. Apply suitable methods to treat or remove outliers and compare the results. **Objectives** • Detect abnormal data points • Improve dataset quality through outlier treatment **Requirements** • Use Boxplot visualization • Apply Z-score or IQR method • Remove or cap outliers • Perform log transformation if necessary **Expected Outcome** • Before and after visualizations • Dataset with treated outliers
4.	**Assignment 4: Data Transformation & Feature Engineering** **Problem Statement** Apply feature transformation and encoding techniques on a dataset to prepare it for machine learning model development. **Objectives** • Normalize and standardize numerical data • Encode categorical variables **Requirements** • Apply Min-Max normalization • Apply Z-score standardization • Perform Label Encoding • Perform One-Hot Encoding **Expected Outcome** • Transformed dataset • Comparison of scaling methods
5.	**Assignment 5: Exploratory Data Analysis (EDA)** **Problem Statement** Perform exploratory data analysis on a dataset using various visualization techniques to identify patterns, relationships, and correlations among variables. **Objectives** • Analyze dataset visually • Interpret statistical relationships **Requirements** • Create Histogram and Pie Chart

	• Create Scatter Plot for bivariate analysis • Generate Heatmap for correlation analysis **Expected Outcome** • Visual graphs with interpretation • Correlation analysis report
6.	**Assignment 6: Simple Linear Regression** **Problem Statement** Build a simple linear regression model to predict student marks based on study hours using Scikit-learn. **Objectives** • Understand regression modeling • Predict continuous values using linear regression **Requirements** • Use dataset containing study hours and marks • Split dataset into training and testing sets • Train Linear Regression model • Plot regression line and predictions **Expected Outcome** • Regression equation • Prediction graph • Predicted output values
7.	**Assignment 7: Multiple Linear Regression** **Problem Statement** Develop a multiple linear regression model using a housing dataset and evaluate the model performance using standard evaluation metrics. **Objectives** • Build regression models with multiple features • Evaluate model accuracy **Requirements** • Use Boston Housing or similar dataset • Train multiple linear regression model • Calculate R^2 Score and RMSE • Analyze feature importance **Expected Outcome** • Performance metrics • Feature contribution analysis
8.	**Assignment 8: Classification using K-Nearest Neighbors (KNN)** **Problem Statement** Implement the K-Nearest Neighbors classification algorithm on a medical dataset and evaluate its classification performance. **Objectives** • Understand instance-based learning • Perform classification using KNN **Requirements** • Use Breast Cancer dataset • Train KNN classifier • Tune different K values • Evaluate using Accuracy, Confusion Matrix, Precision, Recall, and F1-score **Expected Outcome**

	• Optimal K value • Classification performance metrics
9.	**Assignment 9: Decision Tree & Logistic Regression** **Problem Statement** Develop and compare Decision Tree and Logistic Regression models on a classification dataset. Analyze their performance using evaluation metrics and hyperparameter tuning. **Objectives** • Compare classification algorithms • Improve model performance through tuning **Requirements** • Train Decision Tree classifier • Train Logistic Regression classifier • Perform hyperparameter tuning • Compare model accuracy and performance **Expected Outcome** • Comparative performance analysis • Accuracy and evaluation metrics
10.	**Assignment 10: K-Means Clustering** **Problem Statement** Implement K-Means clustering algorithm on a dataset and determine the optimal number of clusters using the Elbow Method. **Objectives** • Understand unsupervised learning • Perform clustering analysis **Requirements** • Apply K-Means clustering algorithm • Use Elbow Method to find optimal K • Visualize clusters graphically **Expected Outcome** • Cluster visualization • Optimal number of clusters

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3 (WK1) (WK2)	2 (WK2)	1 (WK2)	–	2 (WK5)	–	–	–	–	1 (WK9)	–	3	–	–
CO2	3 (WK2) (WK3)	3 (WK3)	2 (WK3) (WK4)	–	2 (WK5)	–	–	–	–	2 (WK9)	1 (WK11)	3	2	–
CO3	3 (WK3) (WK4)	3 (WK4)	3 (WK4)	2 (WK4)	2 (WK5)	–	–	–	1	2 (WK9)	2 (WK10)	3	2	–
CO4	3 (WK3) (WK4)	3 (WK4)	3 (WK4)	2 (WK4)	3 (WK5)	–	–	–	1	2 (WK9)	2 (WK10)	–	3	2

Course Code	:	**1BAIA306C**	**Embedded Systems**	**Semester**	:	**3**
L:T:P	:	**2:0:0**		**Course Type**	:	**PCC**
Hours/Sem. **Teaching**	:	**28Hrs**		**CIE Marks**	:	**50**
Learning (TW+SL)	:	**26Hrs**		**SEE Marks**	:	**50**
Exam	:	**06Hrs**		**Total Marks**	:	**100**
Total Hrs.	:	**60Hrs**		**Credits**	:	**02**

Professional Competency:

Design, develop, interface, and troubleshoot 8051 microcontroller-based embedded systems by programming in Assembly and C, integrating peripherals and motors, and applying systematic testing and debugging techniques for real-world applications.

Course Outcomes:

After completion of the course, student will be able to:

CO1: Understand the architecture and features including registers, memory, timers, I/O ports, and clock system of the 8051 microcontroller**.**

CO2: Develop the basic 8051 assembly programs using addressing modes and instructions.

CO3: Develop 8051 programs in assembly and C for control, I/O, and delay operations and it also includes timer programming.

CO4: Develop software for interfacing peripherals and motors with 8051 microcontroller.

Unit-I	**7 Hrs.**

8051 Architecture: Features of 8051 microcontroller, Internal block diagram, Oscillator and clock, Accumulator, Data pointer, Program counter, Program status word, Stack pointer, Special function registers, Timer/ counter, I/O ports, Memory organization.

Unit-II	**7 Hrs.**

Addressing modes: Immediate, register, direct and indirect addressing modes. **Instruction Set and Programming:** Data transfer, Arithmetic, Control transfer instructions, Logic and compare instructions, and assembly programs.

Unit-III	**7 Hrs.**

8051 Programming in C: Data types and time delay in 8051 C, I/O programming in C, Logical operations in C. **8051 Timer Programming in assembly and C:** programming 8051 timers, counter programming, programming timers 0 and 1 in 8051 C.

Unit-IV	**7 Hrs.**

Interfacing Peripherals with 8051 Microcontroller: LED interfacing, Seven segment LED interfacing, LCD interfacing, Stepper motor interfacing, DC motor interfacing (programs for interfacing peripherals in assembly).

Text Books:

1. Kenneth J. Ayala, "8051 Microcontroller: Architecture, Programming and Applications", 3rd Edition, Thomson publication, 2005.
2. Muhammad Ali Mazidi, Janice Gillespie Mazidi, Rolin D. McKinlay, "The 8051 Microcontroller and Embedded Systems: using Assembly & C", 2nd Edition, Pearson,

2006.

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	**3 (WK1) (WK3) (WK4)**	**1 (WK1) (WK3) (WK4)**	**–**	**–**	**2 (WK4) (WK5) (WK6)**	**–**	**–**	**–**	**–**	**–**	**1 (WK8)**	**2**	**–**	**1**
CO2	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK5**	**–**	**2 (WK2) (WK4) (WK5) (WK6)**	**–**	**–**	**–**	**–**	**–**	**1 (WK8)**	**2**	**1**	**1**
CO3	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**1**	**3 (WK2) (WK4) (WK5) (WK6)**	**–**	**–**	**–**	**–**	**–**	**2 (WK8)**	**2**	**1**	**2**
CO4	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**2 (WK8)**	**3 (WK2) (WK4) (WK5) (WK6)**	**1 (WK1)**	**–**	**2**	**1**	**–**	**2 (WK8)**	**3**	**2**	**2**

Course Code		:	1BAIA306L	**Data Structures Laboratory**	Semester	:	3
L:T:P		:	0:0:2		Course Type	:	PCC
Hours/Sem.	Teaching	:	26 Hrs.		CIE Marks	:	50
					SEE Marks	:	50
	Exam	:	4 Hrs.		Total Marks	:	100
	Total Hrs.	:	30 Hrs.		Credits	:	1

Professional Competency:
The Data Structures Lab equips fresher's with fundamental programming, analytical, and problem-solving skills required to develop efficient software solutions and meet industry expectations in software development
Course Outcomes:
After completion of the course, student will be able to: **CO1:** Analyze and implement various linear and non-linear data structures for problem-solving. **CO2:** Demonstrate the working of stacks, queues, linked lists, trees, and graphs through practical implementation. **CO3:** Apply suitable searching, sorting, hashing, and traversal techniques for efficient data processing. **CO4:** Develop solutions for real-world computational problems by selecting and implementing appropriate data structures.

List of Experiments OR Programs List
1. Develop a Program in C for the following: (a) Declare a calendar as an array of 7 elements (dynamically created array) to represent 7 days of a week. Each element of the array is a structure having three fields: day name (dynamically allocated string), date (integer), and activity description (dynamically allocated string). (b) Write functions create(), read(), and display() to create the calendar, read data from the keyboard, and print weekly activity details report on screen.
2. Develop a Program in C for String operations: (a) Read main string (STR), pattern string (PAT), and replace string (REP). (b) Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions
3. Develop a menu-driven Program in C for Stack of Integers (Array Implementation of Stack with MAX size a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations
4. Develop a Program in C for converting an Infix Expression to Postfix Expression supporting parenthesized and free-parenthesized expressions with operators +, -, *, /, %, ^ and alphanumeric operands.
5. Develop a Program in C for Stack Applications: (a) Evaluation of suffix expression with single-digit operands and operators +, -, *, /, %, ^ (b) Solving Tower of Hanoi problem with n disks.
6. Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations

Course Code	:	1BAIA307C	OOPs with Java	Semester	:	3
L:T:P	:	2:0:2		Course Type	:	IPCC
Hours/Sem Teaching	:	28 +26=54 Hrs		CIE Marks	:	50
Learning (TW+SL)	:	28 Hrs		SEE Marks	:	50
Exam	:	08 Hrs		Total Marks	:	100
Total Hrs.	:	90 Hrs		Credits	:	03

7. Develop a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem,PhNo a. Create a SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack) e. Exit
8. Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
9. Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
10. Develop a Program in C for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method
11. Develop a Program in C for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method
12. Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Professional Competency:
Apply object-oriented programming principles and Java programming concepts to analyze, design, develop, test, and maintain modular, reusable, and efficient software applications using appropriate programming constructs, libraries, and development tools to solve real-world engineering problems.
Course Outcomes:
After completion of the course student will be able to CO1: Explain the core principles of Java including data types, control structures, and object-oriented features like classes and inheritance. CO2: Apply interfaces, packages, and exception handling mechanisms to build modular and robust Java programs. CO3: Develop Java programs involving multithreading and file I/O operations for real-world

problem solving.

CO4: Demonstrate proficiency in Java's Collection Framework and event-driven programming.

UNIT-I	07 Hrs.

Introduction to Java: Characteristics of Java, Types of Java Programs, Command Line Arguments, Java Runtime Environment(JRE), Java Virtual Machine, Java Developer's Kit. **Basic Language Elements:**Structure of a Java Program, Data Types, Java Variables, Java Literal, and Comments in a program, Operators.

UNIT–II	07 Hrs.

Objects and Classes: Defining a Class, Constructors, Multiple Constructors, Inner Classes, Abstract Classes, Wrapper Classes, Conversion of Data Types. **Attributes and Methods:**Attributes, Method Overriding, Object as Parameters, Exception Class, Try and Catch, Multiple catch, Exception Types, Throw and Throws, User Defined Exception.

UNIT–III	07 Hrs.

Inheritance, Interfaces and Packages: Inheritance, Types of Inheritance, Example of Inheritance Implementations, Polymorphism, Employee Example, Inheritance Used in Geometry, Interfaces, Packages. **Multithreading:**Multitasking and Multithreading, The Thread Class, Defining and Running a Thread, Methods of Thread Class.

UNIT–IV	07 Hrs.

Java Collections Framework: Separating Collection Interfaces and Implementation, The Collection Interface, Iterators, Generic Utility Methods. **Interfaces in the Collections Framework. Concrete Collections:** Linked Lists, Array Lists, Hash Sets, Tree Sets, Queues and Dequeues, Priority Queues. **Maps:**Basic Map Operations, Updating Map Entries, Map Views, Weak Hash Maps, Linked Hash Sets and Maps, Enumeration Sets and Maps, Identity Hash Maps. **Views and Wrappers:** Small Collections, Subranges, Unmodifiable Views, Synchronized Views, Checked Views, A Note on Optional Operations.

Text Books:

1. C. Xavier, "Java Programming: A Practical Approach", TMH (Tata McGraw-Hill).
2. Cay S. Horstmann, "Core Java: VOLUME I-FUNDAMENTALS", Pearson, 11thEdition.

Reference Books:

1. Herbert Schildt, "Java: The Complete Reference", McGraw-Hill Education, 11thEdition, 2018.
2. Kathy Sierra and Bert Bates, "Head First Java", O'Reilly Media, 2nd Edition, 2005.
3. E. Balagurusamy, "Programming with Java: A Primer", McGraw Hill, 5th Edition, 2014.
4. Y. Daniel Liang,"Introduction to Java Programming and Data Structures", Pearson, 12th Edition, 2018.
5. Paul Deitel and Harvey Deitel, "Java: How to Program", Pearson Education, 10thEdition, 2014.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes	Programme Outcomes (POs) and WKs											Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3	2	1	–	2	–	–	–	–	–	1	3	2	–

	(WK1) (WK2)	**(WK2)**	**(WK2)**		**(WK5)**						**(WK9)**			
CO2	**3 (WK2) (WK3)**	**3 (WK3)**	**2 (WK3) (WK4)**	**–**	**3 (WK5)**	**–**	**–**	**–**	**–**	**–**	**1 (WK9)**	**3**	**3**	**–**
CO3	**3 (WK3) (WK4)**	**3 (WK4)**	**3 (WK4)**	**2 (WK4)**	**3 (WK5)**	**–**	**–**	**1**	**–**	**1 (WK10)**	**2 (WK9)**	**3**	**3**	**1**
CO4	**2 (WK3) (WK4)**	**2 (WK4)**	**3 (WK4)**	**1 (WK4)**	**3 (WK5)**	**–**	**–**	**1**	**1**	**–**	**2 (WK9)**	**2**	**3**	**1**

Lab Assignments

SL. No.	Assignments
1.	Design a Java class named **Student** with data members such as id, name, and marks. Create methods to: • Store student details • Display all student details • Search a student using ID • Update the marks of a student
2.	Create a Java class **Employee** that demonstrates **constructor overloading** and **method**
3.	Develop a Java program that demonstrates **single inheritance** and **multilevel inheritance** using classes Animal, Dog, and Puppy. • Animal should have a method eat() • Dog should inherit from Animal and include bark() • Puppy should inherit from Dog and include sleep() Write a main class to create an object of Puppy and call all inherited methods.
4.	Write a Java program that implements **multiple inheritance** using **interfaces**. Create two interfaces: • Person with method show()

	• Marks with method displayMarks() Create a class Student that **implements both interfaces** and provides definitions for their methods. Organize the interfaces and class inside a **package named mypack**, and write a test program to create a Student object and call the methods.
5.	Develop a Java program to demonstrate the concepts of **exception handling** using the following: • try-catch-finally block for handling runtime errors • throw keyword to explicitly throw an exception • A **custom exception class** to validate a condition (e.g., invalid age, insufficient balance, negative marks) The program should read user input, handle improper values using exceptions, and ensure that the finally block executes in all cases.
6.	Write a Java program to perform file operations using standard I/O classes: • **Read** data from a text file using FileReader and Scanner • **Write** data to a text file using BufferedWriter
7.	Implement a Java program that demonstrates multithreading using: • A thread created by **extending the** Thread **class** • A thread created by **implementing the** Runnable **interface**
8.	Write a Java program to demonstrate operations on **ArrayList**, **HashMap**, and **TreeSet** for data manipulation.
9.	Design and implement a **simple calculator application** using Java AWT. The calculator should contain the following components: • Labels, TextFields for input • Buttons for operations (+, −, ×, ÷) • A Label/TextField to display the result
10.	Develop a Java AWT or Swing program to demonstrate **event handling** using GUI components such as Buttons and TextFields

4^{th} Semester Scheme and Syllabus

Sl. No.	Course Category	Course Code	Course Title	Teaching-Learning Scheme				Examination Duration (Hrs/Sem.)		Total Hours	Examination Scheme and Evaluation			Credits (C)=Total Hours/30
				L Hrs/Week (Hrs/Sem.)	T Hrs/Week (Hrs/Sem.)	P Hrs/Week (Hrs/Sem.)	Term Work (TW) & Self Learning (SL), (TW+SL) Hrs/Sem.	CIE	SEE		CIE Marks	SEE Marks	Total Marks	
1.	BSC	1BBTA402C	Biology for Engineers	3(42)	0	0	42	3	3	90	50	50	100	3
2.	IPCC	1BAIA403C	Analysis & Design of Algorithms	2(28)	0	2(26)	28	5	3	90	50	50	100	3
3.	PCC	1BAIA404C	Operating Systems	3(42)	0	0	42	3	3	90	50	50	100	3
4.	PCC	1BAIA405C	Machine Learning Algorithms	3(42)	0	0	42	3	3	90	50	50	100	3
5.	IPCC	1BAIA406C	Database Management Systems	2(28)	0	2(26)	28	5	3	90	50	50	100	3
6.	PCC	1BAIA407C	Optimization Techniques	3(42)	0	0	42	3	3	90	50	50	100	3
7.	PCC	1BAIA408L	Machine Learning Laboratory	0	0	2(26)	0	2	2	30	50	50	100	1
8.	Audit	1BAIA409L	Robotics Laboratory	0	0	2(26)	0	2	2	30	50	50	100	1(PP)
9.	HSSM	1BHSA424C	UHV-II	1(14)	0	0	13	2	1	30	50	50	100	1
10.	MC	1BHSB460M / 1BHSC460M / 1BHSA460M	NSS / P.Ed. / Yoga	1	0	0	14	2	–	30	25	–	25	0
			Total	20		8					425	400	825	20

<table>
<tr><td colspan="2">Course Code</td><td>:</td><td>1BBTA402C</td><td rowspan="7">Biology for Engineers</td><td>Semester</td><td>:</td><td>4</td></tr>
<tr><td colspan="2">L:T:P</td><td>:</td><td>3:0:0</td><td>Course Type</td><td>:</td><td>BSC</td></tr>
<tr><td rowspan="5">Hours/Sem.</td><td>Teaching</td><td>:</td><td>42 Hrs</td><td>CIE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Learning (TW+SL)</td><td>:</td><td>42 Hrs</td><td>SEE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Exam</td><td>:</td><td>06 Hrs</td><td>Total Marks</td><td>:</td><td>100</td></tr>
<tr><td>Total Hrs.</td><td>:</td><td>90 Hrs</td><td>Credits</td><td>:</td><td>03</td></tr>
</table>

Course Outcomes:	
After completion of the course student will be able to CO1: Corroborate the concepts of bio mimetics for specific requirements. CO2: Elucidate the basic biological concepts via relevant industrial applications and case studies. CO3: Evaluate the principles of design and development, for exploring novel bioengineering projects. CO4: Think critically towards exploring innovative biobased solutions for ecofriendly and socially relevant problems.	
UNIT-I	**10 Hrs.**
Nature Bioinspired Materials And Mechanisms: Bio inspiration - Introduction, Alliance between Engineering and Biology, Biomimicry - Science mimicking nature. Human Blood substitutes- hemoglobin based oxygen carriers (HBOCs) and perflourocarbons (PFCs). Artificial Intelligence for disease diagnosis. Bioichips & their applications. Biosensors & their applications. Nanobiomolecules in medical science. Biofilms in dental treatment.	
UNIT–II	**10 Hrs.**
Bio Inspiration Models Used In Engineering: BioEcholocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf), Respiration (MFCs), Bird flying (GPS and aircrafts), Lotus leaf effect (Super hydrophobic and self cleaning surfaces), Gecko Feet, Plant burrs (Velcro), Shark skin (Friction reducing swimsuits), Kingfisher beak (Bullet train), Fire fly LED.	
UNIT–III	**10 Hrs.**
Human Organ Systems And Bio Designs: Brain as a CPU system (architecture, CNS and Peripheral Nervous System, Signal transmission, EEG, Robotic arms for prosthetics. Engineering solutions for Parkinson's disease). **Heart** as a pump system (architecture, electrical signalling - ECG monitoring and heart related issues, Reasons for blockages of blood vessels, Design of stents, Pace makers, defibrillators). **Lungs** as purification system gas exchange mechanisms, Spirometry, Ventilators, Heart-lung machine). **Eye** as a Camera system, bionic eye. **Kidney** as a filtration system - dialysis systems. **Muscular and Skeletal Systems** as scaffolds, Bioengineering solutions for muscular dystrophy and osteoporosis.	
UNIT–IV	**10 Hrs.**

Trends In Bioengineering: Bioprinting techniques and materials, 3D printing of ear, Bone and skin, 3D printed foods, Electrical tongue and electrical nose in food science, DNA origami and Biocomputing, Bioimaging and Self healing Bioconcrete (based on bacillus spores, calcium lactate nutrients and biomineralization processes) and Bioremediation and Biomining via microbial surface adsorption (removal of heavy metals like Lead, Cadmium, Mercury, Arsenic), Bio-bleaching.

Reference Books:

1. "Human Physiology", Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
2. "Biology for Engineers", Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012
3. "Biology for Engineers", Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
4. "Biomedical Instrumentation", Leslie Cromwell, Prentice Hall 2011
5. "Biology for Engineers", Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2020.
6. "Biomimetics: Nature-Based Innovation", Yoseph Bar-Cohen, 1st edition, CRC Press, 2012
7. "Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies", D. Floreano and C. Mattiussi, MIT Press, 2008
8. "Bioremediation of heavy metals: bacterial participation", by C R Sunilkumar, N GeethaA C Udayashankar Lambert Academic Publishing, 2019
9. "3D Bioprinting: Fundamentals, Principles and Applications" by Ibrahim Ozbolat, Academic Press, 2016
10. "Electronic Noses and Tongues in Food Science", Maria Rodriguez Mende, Academic Press, 2016

Course Outcomes	Programme Outcomes											Programme Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	PSO 1	PSO 2	PSO 3
CO 1	3 (WK1) (WK2) (WK3) (WK4)	3 (WK1) (WK2) (WK3) (WK4)	3 (WK5)	2 (WK8)	2 (WK2) (WK6)	2 (WK1) (WK5) (WK7)	2 (WK9)	2			3 (WK8)	3	2	2
CO 2	3 (WK1) (WK2) (WK3) (WK4)	3 (WK1) (WK2) (WK3) (WK4)	3 (WK5)	2 (WK8)	3 (WK2) (WK6)	3 (WK1) (WK5) (WK7)	2 (WK9)	2			3 (WK8)	3	2	2
CO 3	3 (WK1) (WK2) (WK3) (WK4)	3 (WK1) (WK2) (WK3) (WK4)	3 (WK5)	2 (WK8)	3 (WK2) (WK6)	2 (WK1) (WK5) (WK7)	3 (WK9)	2			3 (WK8)	3	2	2
CO 4	3 (WK1) (WK2) (WK3) (WK4)	3 (WK1) (WK2) (WK3) (WK4)	3 (WK5)	2 (WK8)	3 (WK2) (WK6)	3 (WK1) (WK5) (WK7)	3 (WK9)	2			3 (WK8)	3	2	2

<table>
<tr><td colspan="2">Course Code</td><td>:</td><td>1BAIA403C</td><td rowspan="6">Analysis & Design of Algorithms</td><td>Semester</td><td>:</td><td>4</td></tr>
<tr><td colspan="2">L:T:P</td><td>:</td><td>2:0:2</td><td>Course Type</td><td>:</td><td>IPCC</td></tr>
<tr><td rowspan="4">Hours/Sem.</td><td>Teaching</td><td>:</td><td>28+26=54Hrs</td><td>CIE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Learning (TW+SL)</td><td>:</td><td>28 Hrs</td><td>SEE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Exam</td><td>:</td><td>08 Hrs</td><td>Total Marks</td><td>:</td><td>100</td></tr>
<tr><td>Total Hrs.</td><td>:</td><td>90 Hrs</td><td>Credits</td><td>:</td><td>03</td></tr>
</table>

Professional Competency:	
Ability to design, analyze, and optimize efficient algorithmic solutions by applying appropriate design paradigms and complexity analysis techniques to solve real-world computational problems under given constraints of time and space.	
Course Outcomes:	
After completion of the course, student will be able to: CO1: Apply asymptotic notations to analyze and compare the time complexity of algorithms. CO2: Design and implement algorithmic solutions using brute-force, decrease-and-conquer, and divide-and-conquer techniques. CO3: Employ transform-and-conquer, space–time tradeoff, and dynamic programming techniques to solve complex computational problems. CO4: Select and apply appropriate algorithmic paradigms such as greedy methods, input enhancement, backtracking, branch and bound, and approximation algorithms to solve real-world problems. CO5: Analyze computational problems and classify them into P, NP, and NP-Complete categories, explaining their inherent limitations.	
UNIT–I	**7 Hrs**
Introduction: Fundamentals of Algorithmic Problem Solving. **Fundamentals Of The Analysis Of Algorithm Efficiency:** Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms. **Brute Force Approaches:** Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching. Exhaustive Search (Travelling Salesman problem and Knapsack Problem). BFS and DFS	
UNIT–II	**7 Hrs**
Decrease and Conquer: Insertion Sort, Topological Sorting. **Divide and Conquer:** Merge Sort, Quick Sort, Binary Tree Traversals, Multiplication of Large Integers and Strassen's Matrix Multiplication. **Transform and Conquer:** Balanced Search Trees, Heaps and Heapsort.	
UNIT-III	**7 Hrs**
Space-Time Tradeoffs: Sorting by Counting, Comparison counting sort, Input Enhancement in String Matching, Horspool's Algorithm. **Dynamic Programming:** Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.	
UNIT-IV	**7 Hrs**
The Greedy Method: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes. **Limitations of Algorithmic Power:** Decision Trees, P, NP, and NP-Complete Problems. **Coping With Limitations of Algorithmic Power**: Backtracking (n-	

Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NP-Hard problems (Knapsack problem).
Text Books:
1. Anany Levitin, “Introduction to the Design & Analysis of Algorithms”, Pearson Education, 3rd Edition, 2017.
Reference Books:
1. Stein, “Introduction to Algorithms”, PHI, 2nd Edition. **2.** Sahni S., Rajasekaran S, Horowitz E, “Computer Algorithms”, Galgotia Publications, 2001.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	**3 (WK1) (WK2)**	**2 (WK1) (WK2)**	-	-	-	-	-	-	-	-	-	**3**	-	-
CO2	**3 (WK1) (WK2) (WK3)**	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	-	-	-	-	-	-	-	-	**3**	-	-
CO3	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**3 (WK6)**	-	-	-	-	-	-	-	**3**	-	-
CO4	**3 (WK2) (WK3) (WK4) (WK5)**	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**3 (WK6)**	**2 (WK6)**	-	-	-	-	-	-	**2**	-	-
CO5	**3 (WK1) (WK2) (WK4)**	**3 (WK1) (WK2) (WK4)**		**3 (WK1) (WK2) (WK3) (WK4)**	-	-	-	-	-	-	-	**2**	-	-

Lab Assignments

S. No	Assignments
1.	**a.** Write a C program to search a given element using binary search method and determine its time complexity. **b.** Write a C program to sort a given set of numbers using the quick sort method and determine its time complexity.
2.	Write a C program to check whether a given graph is connected or not using DFS method and determine its time complexity.
3.	Write a C program to sort a given set of numbers using the merge sort method and determine its time complexity.
4.	Write a C program to print all the nodes reachable from a given starting node in a digraph using BFS method and determine its time complexity.
5.	Write a C program to sort a given set of numbers using the heap sort method and determine its time complexity.
6.	**a.** Write a C program to find the Transitive Closure of a graph using Warshall's algorithm. **b.** Write a C program to find all pair shortest path of a graph using Floyd's algorithm.
7.	Write a C program to implement 0/1 Knapsack problem using Dynamic Programming and determine its time complexity.
8.	Write a C program to find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm and determine its time complexity.
9.	Write a C program to find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm and determine its time complexity.
10.	Write a C program to find the shortest path from a given vertex to other vertices in a weighted connected graph using Dijkstra's algorithm and determine its time complexity.

<table>
<tr><td colspan="2">Course Code</td><td>:</td><td>1BAIA404C</td><td rowspan="7">Operating Systems</td><td>Semester</td><td>:</td><td>4</td></tr>
<tr><td colspan="2">L:T:P</td><td>:</td><td>3:0:0</td><td>Course Type</td><td>:</td><td>PCC</td></tr>
<tr><td rowspan="5">Hours/Sem.</td><td>Teaching</td><td>:</td><td>42 Hrs</td><td>CIE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Learning (TW+SL)</td><td>:</td><td>42 Hrs</td><td>SEE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Exam</td><td>:</td><td>06 Hrs</td><td>Total Marks</td><td>:</td><td>100</td></tr>
<tr><td>Total Hrs.</td><td>:</td><td>90 Hrs</td><td>Credits</td><td>:</td><td>03</td></tr>
</table>

Professional Competency:

Students as an Entry-level developers must apply OS concepts to navigate, manage systems and code

Course Outcomes:

After completion of the course, student will be able to:

CO1: Explain the structure, services, and functions of operating systems.

CO2: Apply process scheduling and synchronization techniques for efficient execution.

CO3: Analyze deadlock conditions and memory management strategies.

CO4: Demonstrate virtual memory concepts and file system implementation methods..

UNIT–I	11 Hrs

Introduction to operating systems, types and services. Role of Operating systems: user view, system view; Operating System structure; Operating System operations; Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines. Process management: Process concept; **Concepts of process:** Process status, Process description, Process model, Operations on processes.

UNIT–II	10 Hrs

Process management, threads and process synchronization. Process Scheduling: Basic concepts; scheduling criteria; Scheduling algorithms; Multiple- Processor scheduling, Inter-process communication (Intd.), **Threads:** concepts, Multi- Threaded Programming: Overview; Multithreading models. **Synchronization:** The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

UNIT-III	10 Hrs

Deadlocks and memory management: Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management Strategies:** Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

UNIT-IV	11 Hrs

Virtual Memory Management: Background; Demand paging; Page replacement; Allocation of frames. **File system: concepts and implementation, secondary storage structures. File System:** File concept; Access methods; Directory structure; File system mounting; File sharing; Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin , Greg Gagne: "Operating System", 7^{th}

Edition, Addison Wesley.
Reference Books:
1. D.M Dhamdhere: “Operating systems - A concept based Approach”, 2nd Edition, Tata McGraw- Hill, 2002..

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	**Programme Outcomes (POs) and WKs**											**Program Specific Outcomes (PSOs)**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**-**	**-**	**-**	**-**	**-**	**1**	**-**	**-**	**1**	**-**	**2**	**-**
CO2	**-**	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**-**	**3 (WK6)**	**1 (WK1) (WK5) (WK7)**	**1 (WK9)**	**1**	**1**	**-**	**1 (WK8)**	**-**	**2**	**-**
CO3	**-**	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**-**	**3 (WK6)**	**1 (WK1) (WK5) (WK7)**	**-**	**1**	**1**	**-**	**1 (WK8)**	**-**	**2**	**-**
CO4	**-**	**2 ((WK1) (WK2) (WK3) (WK4)**	**2 (WK5)**	**-**	**-**	**1 (WK1) (WK5) (WK7)**	**-**	**1**	**1**	**-**	**1 (WK8)**	**-**	**2**	**-**

Course Code	:	1BAIA405C	Machine Learning Algorithms	Semester	:	4
L:T:P	:	3:0:0		Course Type	:	PCC
Hours/Sem. Teaching	:	42 Hrs		CIE Marks	:	50
Learning (TW+SL)	:	42 Hrs		SEE Marks	:	50
Exam	:	06 Hrs		Total Marks	:	100
Total Hrs.	:	90 Hrs		Credits	:	03

Professional Competency:

Develop the ability to design, implement, evaluate, and optimize machine learning models using supervised learning, neural networks, Bayesian learning, instance-based learning, dimensionality reduction, and clustering techniques to solve real-world artificial intelligence and data-driven engineering problems.

Course Outcomes:

After completion of the course, students will be able to:

CO1: Apply ensemble learning techniques, regularization methods, Support Vector Machines, and class imbalance handling to supervised learning problems.

CO2: Design and implement Artificial Neural Network models using perceptron and backpropagation, and evaluate model performance using appropriate metrics and statistical methods.

CO3: Apply Bayesian learning and instance-based learning techniques for classification and regression problems.

CO4: Use dimensionality reduction and clustering techniques to analyze and interpret high-dimensional datasets.

UNIT-I	10 Hrs.

Ensemble Learning: Key concepts of Ensemble Learning: Weak Learners, Strong Learners
Common Types of Ensemble Learning: Bagging (Bootstrap Aggregating), Boosting, Stacking (Stacked Generalization), Voting.

Key Ensemble Learning Algorithms: Random Forest, AdaBoost, Gradient Boosting, XGBoost (Extreme Gradient Boosting). Advantages and Disadvantages of Ensemble Learning, Applications of Ensemble Learning.

Regularization Techniques: Ridge and Lasso Regression Techniques.

Key concepts of SVM: Hyper-plane, Support Vectors, Margin, Linear SVM, Non-linear SVM. Working of SVM, Soft margin of SVM, Advantages and disadvantages of SVM, Applications of SVM.

Handling Class Imbalance: Re-sampling Techniques (SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN, Cost-sensitive learning).

UNIT-II	11 Hrs.

Artificial Neural Networks (ANN): Introduction, Neural Network Representations, Appropriate Problems For Neural Network Learning, Perceptron, Multilayer Networks And The Back propagation Algorithm, Remarks On The Back propagation Algorithm, An Illustrative Example: Face Recognition.

Hypothesis and Performance Evaluation: Basic Performance Criterion, Precision and recall, Other ways to measure Performance, Estimating Hypothesis Accuracy, Basics of Sampling Theory, General approach for deriving confidence intervals, difference in error of two hypothesis, comparing learning algorithms.

UNIT-III	10 Hrs.

Bayesian learning: Introduction, Bay's theorem, Maximum likelihood and least squared

hypothesis, Maximum likelihood hypothesis for predicting probabilities, Minimum Description length principle, Bay's optimal classifier, Gibbs algorithm, Naive Bay's Classifier. An Example: Classify Text. **Instance Based Learning**: Introduction, k-Nearest Neighbor Learning, Locally Weighted Regression, Radial Basis function, and case based reasoning.	
UNIT-IV	**11 Hrs.**
Dimensionality Reduction: Introduction, Subset Selection, Principal Components Analysis, Factor Analysis, Multi dimensional scaling, Linear descreminant analysis, isomap, Locally Linear Embedding. **Clustering**: Introduction, Mixture Densities, K-means Clustering, Expectation Maximization Algorithm, Mixture Latent Variable models, Supervised learning after clustering, Hierarchical clustering, Choosing the number of clusters	
Textbooks:	
1. Tom Mitchell, Machine Learning, McGraw- Hill Publications, 2nd Edition, 2013. 2. Ethem Alpaydin, Introduction to Machine Learning, MIT press, Cambridge, Massachusetts, London, 2nd Edition, 2010.	
Reference Books:	
1. Trevor Hastie. Robert Tipeshirani, Jerome Fredman, Elements of Statistical Learning, Springer, 2nd Edition, 2010. **2.** Luis Pedro Coelho and Willi Richart, Building Machine Learning Systems with Python, PACKT Publication, 2nd Edition, 2013.	

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	**Programme Outcomes (POs) and WKs**											**Program Specific Outcomes**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO 1	**3 (WK2) (WK3)**	**3 (WK3)**	**3 (WK3) (WK4)**	**2 (WK4)**	**3 (WK5)**	–	–	–	–	**2 (WK9)**	**1 (WK11)**	**3**	**3**	–
CO 2	**3 (WK3) (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK5)**	–	–	–	**1 (WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**3**	**2**
CO 3.	**3 (WK3) (WK4)**	**3 (WK4)**	**3 (WK4)**	**2 (WK4)**	**2 (WK5)**	–	–	–	**1 (WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**2**	**2**
CO 4	**3 (WK4) (WK5)**	**3 (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK5)**	–	–	–	**2 WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**3**	**2**

Course Code	:	1BAIA406C	Database Management System	Semester	:	4
L:T:P	:	2:0:2		Course Type	:	IPCC
Hours/Sem. Teaching	:	28+26=54Hrs		CIE Marks	:	50
Hours/Sem. Learning (TW+SL)	:	28Hrs		SEE Marks	:	50
Hours/Sem. Exam	:	08 Hrs		Total Marks	:	100
Hours/Sem. Total Hrs.	:	90 Hrs		Credits	:	03

Professional Competency:

Develop the ability to analyze, design, implement, and manage database systems using relational and NoSQL technologies, ensuring data integrity, security, consistency, and efficient transaction processing for real-world applications.

Course Outcomes:

After completion of the course, student will be able to:

CO1: Explain database concepts, architectures, data models, and design conceptual database schemas using ER modeling techniques.

CO2: Apply relational database design principles, relational algebra operations, ER-to relational mapping, and normalization techniques to develop efficient database schemas.

CO3: Construct SQL queries for database creation, manipulation, retrieval, views, constraints, and transaction management in relational database systems.

CO4: Analyze transaction processing and concurrency control mechanisms, and evaluate NoSQL databases and Big Data storage systems for modern data-intensive applications.

UNIT–I	07 Hrs.

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.

Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. **Conceptual Data Modelling using Entities and Relationships:** Entity types, Entity sets and structural constraints, Weak entity types, ER diagrams,Specialization and Generalization.

UNIT–II	07 Hrs.

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.

Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra.

Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping.

Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

UNIT-III	07 Hrs.
SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL **SQL: Advanced Queries:** More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL. **Transaction Processing:** Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL.	
UNIT-IV	**07 Hrs.**
Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. **NOSQL Databases and Big Data Storage Systems:** Definition of NOSQL, Types of NOSQL, Advantages of NOSQL, Difference between SQL and NOSQL, CAP Theorem NOSQL Tools, Basic NOSQL CURD Operations, and What is MongoDB, Advantages of MongoDB, Data types in MongoDB, CURD Operations, NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j.	
Text Books:	
1. Elmasri, R., & Navathe, S. B. (2017). Fundamentals of Database Systems (7th ed.). Pearson. 2. Ramakrishnan, R., & Gehrke, J. (2014). Database Management Systems (3rd ed.). McGraw-Hill Education. 3. Acharya, S., & Chellappan, S. (2015). Big Data and Analytics. 1st ed. Wiley India.	
Reference Books:	
1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw-Hill Education, 7th Edition, 2019. 2. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The Complete Book", Pearson Education, 2nd Edition. 3. Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, "Modern Database Management", Pearson, 13th Edition, 2019. 4. Ivan Bayross, "SQL, PL/SQL: The Programming Language of Oracle", BPB Publications. 5. Shannon Bradshaw, Eoin Brazil, Kristina Chodorow, "MongoDB: The Definitive Guide", O'Reilly Media, 3rd Edition. 6. Eric Redmond, Jim R. Wilson, "Seven Databases in Seven Weeks", Pragmatic Bookshelf, 2nd Edition.	

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK5)**	**1 (WK8)**	**1 (WK2) (WK6)**	**–**	**–**	**–**	**1**	**–**	**1 (WK8)**	**2**	**1**	**1**
CO2	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK1) (WK2) (WK3) (WK4)**	**3 (WK5)**	**1 (WK8)**	**2 (WK2) (WK6)**	**–**	**–**	**–**	**1**	**–**	**1 (WK8)**	**3**	**2**	**2**
CO3	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK1) (WK2) (WK3) (WK4)**	**2 (WK5)**	**1 (WK8)**	**3 (WK2) (WK6)**	**–**	**–**	**1**	**2**	**1**	**1 (WK8)**	**2**	**3**	**3**
CO4	**2 (WK1) (WK2) (WK3) (WK4)**	**3 (WK1) (WK2) (WK3) (WK4)**	**2 (WK5)**	**3 (WK8)**	**3 (WK2) (WK6)**	**1 (WK1) (WK5) (WK7)**	**1 (WK9)**	**1**	**1**	**1**	**2 (WK8)**	**2**	**3**	**3**

Lab Assignments

Sl. No	**Assignments**
1	Create a table called Employee & execute the following. Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION) 1. Create a user and grant all permissions to theuser. 2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2	Create a table called Employee that contain attributes EMPNO,ENAME, JOB, MGR,SAL & execute the following. 1. Add a column commission with domain to the Employeetable. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.
3	Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Orderby. Employee(E_id, E_name, Age, Salary)

	1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employeetable 3. Find the Maximum age from employee table. 4. Find the Minimum age from employeetable. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)
5	Create cursor for Employee table & extract the values from the table. Declare the variables ,Open the cursor & extrct the values from the cursor. Close the cursor. Employee(E_id, E_name, Age, Salary)
6	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.
7	Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

Course Code		:	1BAIA407C	Optimization Technique	Semester	:	4
L:T:P		:	3:0:0		Course Type	:	PCC
Hours/Sem.	Teaching	:	42 Hrs		CIE Marks	:	50
	Learning (TW+SL)	:	42 Hrs		SEE Marks	:	50
	Exam	:	06 Hrs		Total Marks	:	100
	Total Hrs.	:	90 Hrs		Credits	:	03

Professional Competency:

Apply classical, stochastic, and evolutionary optimization techniques to formulate, optimize, and evaluate machine learning models, enabling efficient problem-solving, improved model performance, and informed decision-making in real-world AI and data-driven applications.

Course Outcomes:

After completion of the course, student will be able to:

CO1: Grasp essential concepts in function optimization and Connect Optimization with Machine Learning

CO2: Develop the skill to pick the right optimization algorithm based on the problem

CO3: Create visualizations for function optimization

CO4: Apply optimization techniques to ML based real-world problems

UNIT–I	10 Hrs.

Foundation: Function Optimization, Candidate solutions, Objective functions, Evaluation costs. **Optimization and Machine Learning**: Introduction to ML and Optimization, Learning as optimization, Optimization in ML project. **How to Choose an Optimization Algorithm**: Optimizing algorithms, Differentiable objective function, Non differentiable objective function. **Background:** No Free Lunch Theorem for Machine Learning, Implications for optimization and Machine learning. **Local Optimization vs. Global Optimization**: Local Optimization, Global Optimization, Local Optimization vs. Global Optimization. **Premature Convergence:** Convergence in ML, Premature convergence, Addressing premature convergence. **Creating Visualization for Function Optimization**: Visualization for function optimization, Visualize 1D function optimization, **Visualize 2D function optimization**, **Stochastic Optimization Algorithms**: Stochastic optimization and algorithms, Practical considerations for Stochastic Optimization. **Random Search and Grid Search:** Naïve function optimization algorithms, Random search for function optimization, Grid search for function optimization.

UNIT–II	11 Hrs.

Local Optimization: Gradient in Machine Learning, Derivative and gradient, Worked examples of calculating derivatives, Interpreting derivatives, Calculating derivative of a function. **Univariate Function Optimization**: Univariate function optimization, Convex univariate function optimization, Non convex univariate function optimization. **Pattern Search**: Nelder-Mead Optimization Algorithm, Nelder-Mead example in Python, Nelder-Mead on challenging functions. **Second Order optimization algorithms**: The BFGS and L-BFGS-B Optimization Algorithms, Worked examples of BFGS. **Stochastic Hill Climbing algorithms:** Stochastic Hill Climbing algorithm and its implementation, Examples of applying Stochastic Hill Climbing algorithms. **Iterated Local Search:** Introduction to iterative local search, Ackley objective function, Stochastic Hill Climbing algorithm with random restarts, Iterated local search algorithms.

UNIT-III	10 Hrs.

Global Optimization: Simple Genetic Algorithm: Genetic algorithm from scratch, genetic algorithm for Onemax, Genetic algorithm for function optimization. **Evolution Strategies**: Develop a (μ,λ)-ES, develop $(\mu+\lambda)$-ES. **Differential Evolution:** Differential evolution algorithm from scratch, Differential evolution algorithm on the sphere function. **Simulated Annealing**: Implement simulated annealing and worked Example.

UNIT-IV	11 Hrs.

Gradient Descent: Gradient Descent Optimization: Gradient descent and worked example. Gradient descent optimization, Gradient descent with momentum and its visualization. Gradient Descent with AdaGrad,Gradient Descent with RMSProp,Gradient Descent with Adadelta, Adam Optimization Algorithm

Projects:

Use Optimization Algorithms to Manually Fit Regression Models: Optimize linear and logistic regression models,

Optimize Neural Network Models: Optimize a perceptron and a multi layer perceptron.

Feature Selection using Stochastic Optimization: Optimization for feature selection, Enumerate all feature subsets.

Manually Optimize Machine Learning Model: Mannual hyper parameter optimization, Perceptron hyper parameter optimization, XGBOOST hyper parameter optimization.

Text Books:

1. Jayson Brownlee, Optimization Techniques for Machine Learning, Machine learning mastery, 2021.

Reference Books:

1. Strang, G. (2019). Linear Algebra and Learning from Data. Wellesley-Cambridge Press.
2. Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.
3. Sra, S., Nowozin, S., & Wright, S. J. (Eds.). (2012). Optimization for Machine Learning. MIT Press.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3 (WK1) (WK2) (WK3)	2 (WK1) (WK2) (WK3)	–	–	1 (WK2)	–	–	–	–	–	1 (WK8)	3	2	–
CO2	2 (WK2) (WK3) (WK4)	3 (WK2) (WK3) (WK4)	1 (WK5)	2 (WK8)	3 (WK2) (WK6)	–	–	–	–	–	1 (WK8)	2	3	–
CO3	1 (WK2)	2 (WK2) (WK3)	–	3 (WK8)	3 (WK2) (WK6)	–	–	–	–	–	1 (WK8)	–	2	–
CO4	2 (WK2) (WK3) (WK4)	2 (WK2) (WK3) (WK4)	3 (WK5)	3 (WK8)	3 (WK2) (WK6)	1 (WK5) (WK7)	–	–	–	–	2 (WK8)	2	3	2

<table>
<tr><td colspan="2">Course Code</td><td>:</td><td>1BAIA408L</td><td rowspan="6">Machine Learning Lab</td><td>Semester</td><td>:</td><td>4</td></tr>
<tr><td colspan="2">L:T:P*</td><td>:</td><td>0:0:2</td><td>Course Type</td><td>:</td><td>PCC</td></tr>
<tr><td rowspan="4">Hours/Sem.</td><td>Teaching</td><td>:</td><td>26 Hrs.</td><td>CIE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Learning (TW+SL)</td><td>:</td><td>00 Hrs.</td><td>SEE Marks</td><td>:</td><td>50</td></tr>
<tr><td>Exam</td><td>:</td><td>04 Hrs.</td><td>Total Marks</td><td>:</td><td>100</td></tr>
<tr><td>Total Hrs.</td><td>:</td><td>30 Hrs.</td><td>Credits</td><td>:</td><td>01</td></tr>
</table>

<table>
<tr><td colspan="2">Professional Competency:</td></tr>
<tr><td colspan="2">Develop the ability to design, implement, evaluate, and optimize machine learning algorithms using supervised learning, neural networks, Bayesian learning, instance-based learning, dimensionality reduction, clustering, and ensemble techniques to solve complex real-world artificial intelligence and data-driven engineering problems.</td></tr>
<tr><td colspan="2">Course Outcomes:</td></tr>
<tr><td colspan="2">After completion of the course, students will be able to:
CO1: Apply supervised learning techniques such as Support Vector Machines (SVM), ensemble learning algorithms, and regularization methods to solve classification and prediction problems on real-world datasets.
CO2: Design and implement Artificial Neural Network (ANN), Bayesian learning, and instance-based learning models for classification and regression tasks, and evaluate their performance using appropriate metrics.
CO3: Apply dimensionality reduction and clustering techniques, including PCA, Kernel PCA, LDA, K-Means, and DBSCAN, to analyze, preprocess, and interpret high-dimensional data.
CO4: Develop and compare hybrid machine learning solutions by integrating feature engineering, dimensionality reduction, clustering, ensemble learning, and neural network techniques to solve real-world AI applications such as image classification, text classification, customer segmentation, anomaly detection, and sentiment analysis.</td></tr>
<tr><td>SL. NO.</td><td>List of Assignments</td></tr>
<tr><td>1.</td><td>PCA + SVM for Image Classification

Problem Statement:

Design a system that uses Principal Component Analysis (PCA) for dimensionality reduction and Support Vector Machine (SVM) for classification of high-dimensional image data.

Objectives:
• Reduce image dimensionality using PCA.
• Train an SVM classifier on reduced feature vectors.
• Evaluate classification accuracy and computational efficiency.

Dataset: MNIST Handwritten Digits Dataset (Kaggle/UCI)</td></tr>
<tr><td>2.</td><td>Clustering + ANN for Customer Behavior Prediction</td></tr>
</table>

	Problem Statement: Cluster customers into meaningful groups using **K-Means Clustering** and use an **Artificial Neural Network (ANN)** to predict future purchase behavior. **Objectives:** • Perform customer segmentation using clustering. • Train an ANN model for customer behavior prediction. • Compare prediction performance with and without clustering. **Dataset:** Mall Customer Segmentation Dataset (Kaggle)
3.	**Feature Reduction → Ensemble Learning Pipeline** **Problem Statement:** Develop a machine learning pipeline using **PCA/LDA** for feature reduction followed by **Random Forest** and **Gradient Boosting** classifiers. **Objectives:** • Reduce feature dimensionality. • Train ensemble learning models. • Compare classification performance before and after feature reduction. **Dataset:** Breast Cancer Wisconsin Dataset (UCI)
4.	**Real-Time Anomaly Detection (Clustering + Ensemble)** **Problem Statement:** Develop an anomaly detection system using **DBSCAN clustering** and **Isolation Forest** for identifying abnormal data instances. **Objectives:** • Detect outliers using density-based clustering. • Improve anomaly detection using ensemble learning. • Evaluate fraud detection performance. **Dataset:** Credit Card Fraud Detection Dataset (Kaggle)
5.	**Comparative Study: SVM vs ANN vs Ensemble vs Bayesian** **Problem Statement:** Implement and compare **Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes, and Ensemble Learning** algorithms on the same dataset.

	Objectives: • Compare multiple machine learning models. • Evaluate using Accuracy, Precision, Recall, F1-Score, and Training Time. • Identify the best-performing algorithm. **Dataset:** Iris Dataset (UCI) or Wine Dataset (UCI)
6.	**Kernel PCA + SVM for Nonlinear Data** **Problem Statement:** Apply **Kernel PCA** for nonlinear feature extraction and classify data using **Support Vector Machine (SVM).** **Objectives:** • Perform nonlinear dimensionality reduction. • Train SVM on transformed features. • Compare results with conventional PCA. **Dataset:** Two Spirals Dataset (Synthetic) / Make_Moons Dataset (Scikit-learn)
7.	**Clustering-Based Preprocessing for ANN** **Problem Statement:** Use **K-Means Clustering** as a preprocessing step to improve the classification performance of an **Artificial Neural Network (ANN).** **Objectives:** • Cluster input data before training. • Train ANN using clustered data. • Evaluate performance improvement. **Dataset:** Human Activity Recognition (UCI HAR) Dataset
8.	**Ensemble + Feature Selection for Text Classification** **Problem Statement:** Develop a text classification system using **TF-IDF**, **feature selection**, and **ensemble learning algorithms** for sentiment analysis. **Objectives:** • Extract textual features using TF-IDF. • Perform feature selection. • Train and compare ensemble classifiers.

	Dataset: IMDB Movie Reviews Dataset
9.	**Hybrid Model: Bayesian + Ensemble Learning** **Problem Statement:** Develop a hybrid classification model by combining **Naïve Bayes** with **ensemble learning techniques** to improve probabilistic predictions. **Objectives:** • Implement Bayesian classification. • Integrate ensemble methods for enhanced prediction. • Evaluate spam detection performance. **Dataset:** SMS Spam Collection Dataset (UCI)
10.	**Dimensionality Reduction Impact Study** **Problem Statement:** Analyze the impact of different dimensionality reduction techniques (**PCA, t-SNE, and UMAP**) on the performance of **SVM** and **ANN** classifiers. **Objectives:** • Apply multiple dimensionality reduction techniques. • Compare classifier performance. • Analyze computational efficiency and visualization quality. **Dataset:** Fashion-MNIST Dataset

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	**Programme Outcomes (POs) and WKs**											**Program Specific Outcomes**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	**3 (WK2) (WK3)**	**3 (WK3)**	**3 (WK3) (WK4)**	**2 (WK4)**	**3 (WK5)**	**–**	**–**	**–**	**–**	**2 (WK9)**	**1 (WK11)**	**3**	**3**	**–**
CO2	**3 (WK3) (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK5)**	**–**	**–**	**–**	**1 (WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**3**	**2**
CO3	**3 (WK4) (WK5)**	**3 (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK5)**	**–**	**–**	**–**	**1 (WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**3**	**2**
CO4	**3 (WK4) (WK5)**	**3 (WK4)**	**3 (WK4)**	**3 (WK4)**	**3 (WK5)**	**–**	**–**	**–**	**2 (WK8)**	**2 (WK9)**	**2 (WK10)**	**3**	**3**	**3**

Course Code	:	1BAIA409L	Robotics Lab.	Semester	:	4
L:T:P*	:	0:0:2		Course Type	:	Audit
Hours/Sem. Teaching	:	26 Hrs.		CIE Marks	:	50
Learning (TW+SL)	:	00 Hrs.		SEE Marks	:	50
Exam	:	04 Hrs.		Total Marks	:	100
Total Hrs.	:	30 Hrs.		Credits	:	01

Professional Competency:

Develop the ability to program, analyze, simulate, and deploy industrial robotic systems by integrating robot kinematics, motion planning, artificial intelligence, machine vision, autonomous navigation, and robotic simulation tools to solve real-world industrial automation and smart manufacturing problems.

Course Outcomes:

After completion of the course, students will be able to:

CO1: Apply robot programming concepts, coordinate transformations, kinematics, and motion planning to perform industrial robotic operations such as point-to-point motion, linear motion, pick-and-place, and trajectory generation using articulated and SCARA robots.

CO2: Design and implement AI-enabled robotic applications by integrating machine vision, object detection, image processing, and machine learning techniques for intelligent object sorting, conveyor tracking, assembly, and color-based automation.

CO3: Develop and evaluate autonomous robotic systems using Automated Mobile Robots (AMRs) for navigation, obstacle avoidance, warehouse logistics, and real-time decision-making in dynamic environments.

CO4: Model, simulate, and analyze robotic systems using RoboAnalyzer by applying Denavit-Hartenberg (DH) parameters, workspace analysis, trajectory planning, collision detection, and robot dynamics for industrial robotic applications.

SL. NO.	Lab. Assignments
1.	**Mitsubishi 6-DOF Articulated Robot (8 Kg Payload) Experiments** **Robot Coordinate Systems and Motion Programming** **Objective:** Understand joint space and Cartesian space movements. **Activities:** • Teach robot positions. • Execute Point-to-Point (PTP) motion. • Execute Linear motion.

	Outcome: • Understand robot kinematics. • Coordinate transformations.
2.	**Pick and Place Operation** **Objective:** Program robotic manipulation tasks. **Activities:** • Pick objects from source location. • Place at destination. **AI Extension:** • Vision-based object identification.
3.	**AI-Based Object Sorting** **Objective:** Integrate AI vision with robot. **Tasks:** • Detect object color/shape using camera. • Classify objects using ML. • Robot sorts objects automatically.
4.	**Name Tracing** **Objectives** 1. To understand the basic programming and operation of a 6-DOF articulated industrial robot. 2. To learn robot path planning and trajectory generation for drawing alphanumeric characters. 3. To convert text input (student name or predefined text) into robot motion commands. 4. To study the relationship between Cartesian coordinates and robot end-effector movement. 5. To develop robot programs for tracing letters, numbers, and simple geometric shapes on a writing surface. 6. To analyze the accuracy, repeatability, and smoothness of robot movements during tracing operations. 7. To understand the application of industrial robots in automated marking, engraving, painting, and writing tasks.
5.	**Color Sorting**

	Objectives 1. To understand the integration of machine vision and industrial robotics for automated color-based sorting. 2. To identify and classify objects based on their colors using a camera and image processing techniques. 3. To develop a robot program for pick-and-place operations based on color recognition. 4. To learn the working principles of computer vision in robotic automation systems. 5. To perform object detection and color classification using AI/image processing algorithms. 6. To understand coordinate mapping between the vision system and robot workspace. 7. To automate the segregation of colored objects into designated bins or locations. 8. To evaluate the accuracy and efficiency of robotic color sorting operations. 9. To study the applications of AI-enabled robotic systems in smart manufacturing and quality control. 10. To demonstrate real-time decision-making by a robot based on visual input data.
6.	**Mitsubishi SCARA Robot Experiments** **High-Speed Pick and Place** **Applications:** • Electronics assembly • Packaging **Measurements:** • Cycle time • Repeatability
7.	**PCB Component Placement Simulation** **Objective:** Simulate electronic assembly tasks. **Activities:** • Component identification. • Placement accuracy analysis.
8.	**Conveyor Tracking System** **Objective:**

	Track moving objects on conveyor. **AI Integration:** • Object detection • Predictive positioning
9.	**Vision Guided Assembly** **Tasks:** • Detect component orientation. • Automatic assembly using SCARA.
10	**Robot Modeling using DH Parameters** **Objective:** Create robot models.

Table: Matrix to describe the mapping of COs with POs (considering WKs) and PSOs

Course Outcomes (COs)	Programme Outcomes (POs) and WKs											Program Specific Outcomes		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	3 (WK3) (WK4)	3 (WK4)	3 (WK4)	3 (WK4)	3 (WK5)	–	–	–	1 (WK8)	2 (WK9)	2 (WK10)	3	3	2
CO2	3 (WK3) (WK4)	3 (WK4)	3 (WK4)	3 (WK4)	3 (WK5)	–	–	–	2 (WK8)	2 (WK9)	2 (WK10)	3	3	3
CO3	3 (WK4) (WK5)	3 (WK4)	3 (WK4)	3 (WK4)	3 (WK5)	2 (WK6)	1 (WK7)	2 (WK8)	2 (WK9)	2 (WK10)	2 (WK11)	3	3	3
CO4	3 (WK3) (WK4)	3 (WK4)	3 (WK4)	3 (WK4)	3 (WK5)	2 (WK6)	–	1 (WK8)	1 (WK9)	2 (WK10)	2 (WK11)	3	3	2