Curriculum and Syllabus

For

MASTER OF COMPUTER APPLICATIONS (MCA)

Under

Choice Based Credit System (CBCS)

[With effect from the Session 2024-25]

USHA MARTIN UNIVERSITY
Ranchi, Jharkhand

(Recognized by UGC under Sec. 2(f) of UGC Act 1956)

Semester Wise Course Distribution

Sl No.	Course Code	Title of the Course	Lecture Hours/ Weeks			Total Credits
			L	T	P	
SEMESTER – I						
1	MCA-CC101	Operating System	3	0	0	03
2	MCA-CC104	Programming in Java	3	1	0	04
3	MCA-CC106	Programming and Data Structure	3	0	0	03
4	MCA-CC107	Discrete Mathematics	3	0	0	03
5	MCA-CC108	Data Analytics	3	0	0	03
6	MCA-CP102	Programming in Java Lab	0	0	4	02
7	MCA-CP104	Programming and Data Structures Lab	0	0	4	02
8	MCA-CP105	Data Analytics Lab	0	0	4	02
Credits- Sem I						**22**

SEMESTER – II						
1	MCA-CC201	Software Engineering	3	1	0	04
2	MCA-CC203	Database Management System	3	0	0	03
3	MCA-CC204	Operation Research and Optimization Techniques	3	1	0	04
4	MCA-CC206	Artificial Intelligence & Machine Learning	3	0	0	03
5	MCA-CC207	Computer Networks	3	0	0	03
6	MCA-CP201	Data Communication and Computer Networks Lab	0	0	4	02
7	MCA-CP204	Artificial Intelligence & Machine Learning Lab	0	0	4	02
8	MCA-CP205	Database Management System Lab	0	0	4	02
Credits- Sem II						**23**

SEMESTER – III						
1		Elective1 (Set A)	3	1	0	04
2		Elective2 (Set A)	3	1	0	04
3		Elective3 (Set A)	3	1	0	04
4		Elective4 (Set B)	3	0	0	03
5	MCA-CC301	Values and Ethics of Profession	3	0	0	03
6	MCA-CC302	Accountancy and Financial Management	3	0	0	03
7	MCA-CP302	Accountancy Lab	0	0	4	02
8	MCA-CP303	Seminar and Technical Writing				02
9		Elective Lab1 (Set B)	0	0	4	02
Credits- Sem III						**27**

SEMESTER – IV						
1	MCA-CP402	Major Project and Dissertation				12
2	MCA-CP403	Comprehensive Viva				2
Credits- Sem IV						**14**
Total Credits (Sem I+Sem II+Sem III+Sem IV)						**86**

List of Elective Courses (Set A)

Sl No.	Elective Course Code (EC)	Course Title
1	MCA-ECA301	Cryptography
2	MCA-ECA302	Compiler Design
3	MCA-ECA306	Parallel Computing
4	MCA-ECA307	Cloud Computing
5	MCA-ECA308	Bio-Informatics
6	MCA-ECA309	Internet of Things
7	MCA-ECA310	Theory of Computation
8	MCA-ECA311	Data Mining and Data Warehousing
9	MCA-ECA312	Deep Learning

List of Elective Courses (Set B)

Sl No.	Elective Course (EC)		Sl No.	Elective Practical (EP)	
	Code	Title		Code	Title
1	MCA-ECB301	Mobile Computing	1	MCA-EPB301	Mobile Computing Lab
2	MCA-ECB302	Windows Programming With Visual Basic.Net	2	MCA-EPB302	Windows Programming With Visual Basic.Net Lab
3	MCA-ECB304	System Administration and Linux	3	MCA-EPB304	System Administration and Linux Lab
4	MCA-ECB305	Image Processing	4	MCA-EPB305	Image Processing Lab
5	MCA-ECB306	Advanced Python Programming	5	MCA-EPB306	Advanced Python Programming Lab

Naming Conventions

1. **CC –** Core Course
2. **CP –** Core Pracical
3. **ECA –** Elective Course Set A
4. **ECB –** Elective Course Set B
5. **EPB –** Elective Practical Set B

SEMESTER - I

MCA-CC101 Operating System Cr 3

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

MCA-CC101.CO1	Explain the difference between different types of modern operating systems, virtual machines and their structure of implementation and applications.
MCA-CC101.CO2	Discuss the difference between process & thread, issues of scheduling of user-level processes /threads and their issues & use of locks, semaphores, monitors for synchronizing multiprogramming with multithreaded systems and implement them in multithreaded programs.
MCA-CC101.CO3	Understand and Implement the concepts of deadlock in operating systems and how they can be managed / avoided and implement them in multiprogramming system.
MCA-CC101.CO4	Explain the design and management concepts along with issues and challenges of main memory, virtual memory and file system.
MCA-CC101.CO5	Manage the types of I/O management, disk scheduling, protection and security problems faced by operating systems and how to minimize these problems.

Unit I. Introduction: OS and the Computer System, Importance of OS, Basic concepts and terminology, Types of OS: Batch Processing Systems, Multiprogramming Systems, Time Sharing Systems, Real Time Operating Systems, Distributed Operating Systems, Modern Operating Systems, Multitasking, Timesharing, Multithreading, Multiprogramming.

Unit II. Design and implementation of OS Process: Concept and views of process, OS view of processes, Process Control Block, Threads, Scheduling algorithms, performance evaluation

Unit III. Inter Process Communication and Synchronization: Inter process communication and synchronization, mutual exclusion, semaphores, classical problem for concurrent programming,critical region and conditional critical region, monitors, messages.
Deadlock: Deadlock criteria, prevention, avoidance, detection and recovery algorithms.

Unit IV. Memory Management: Resource manager, file management, processor management, device management, Memory Management-paging, swapping, design issues for paging system, segmentation, Virtual memory concept, demand paging, page replacement algorithm.

Unit V. File System, Protection & Security: Security and protection, policies and mechanism, authentication, protection and access control, formal models of protection. File systems, security and protection mechanism, Input/out systems, processes and processors in distributed system. Performance measurement, monitoring and evaluation. Multiprocessor system, classification and types, OS function and requirements.

Text Book and References:

1. Operating System Concepts, A. Silverschwatz, P. Galvin & G.Gange , Willey
2. Operating System Concepts, Milenekovic, McGraw Hill
3. Modern Operating Systems, A.S. Tanenbaum, Pearson Education
4. Operating Systems & Systems Programming, P. Balakrishna Prasad, Scitech Publications Pvt. Ltd.

MCA-CC104 Programming in Java Cr 4

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

MCA-CC104.CO1	Identify basic Java programming constructs, including data types, variables, operators, and control structures.
MCA-CC104.CO2	Explain the principles of Object-Oriented Programming (OOP) concepts such as encapsulation, inheritance, and polymorphism in Java.
MCA-CC104.CO3	Develop Java programs using classes, interfaces, and packages to solve real-world problems.
MCA-CC104.CO4	Analyze and debug Java applications to identify logical, syntax, and runtime errors.
MCA-CC104.CO5	Design and implement advanced Java applications integrating concepts such as multithreading, exception handling, and AWT programming.

Unit I. Introduction to Java

Features of Java, Java Program Structure, Understanding the semantic and syntax differences between C++ and Java, Java Tokens, Java Virtual Machine (Bytecodes), Compiling and Executing a Java Program, Variables, Constants, Data Types, Scope of Variables, Type Casting, Operators, Expressions, Decision Making and Branching, Looping (While, Do, For, Jumps in Loops, Labelled Loops).

Arrays, Strings and I/O

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically,Java Strings: The Java String class, Methods of String class, Manipulating Strings, String Buffer Classes, Methods of String Buffer class, String Tokenizer class, the Scanner class.

Unit II. Classes, Objects and Methods

Class, Object, Constructors, Method Overloading, Inheritance, Method Overriding, This and Super, Final Variables and Methods, Final Classes, Finalize method, Abstract Methods and Classes, Visibility Control

Unit III. Interfaces and Packages

Defining, Extending and Implementing Interfaces, Java API packages, Using Standard Java Packages (util, lang, io, net), Creating a package, Accessing and using a Package.

Unit IV. Multithreading and Exception Handling

Creating Threads, Extending the Thread Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Thread Priority, Thread Synchronization, Runnable Interface Types of Errors, Exceptions, Built-in exceptions, Exception handling code (Try, Catch and Finally), Throwing our own Exception.

Unit V. Applets and AWT Programming

Applet : Java Applets, Applet Life Cycle, Passing Parameters to Applets, Applet Tag, Embedding applets to HTML file, Running the applet

AWT(Abstract Window Toolkit)

Introduction to AWT, Graphics classes and its methods, Drawing Lines, rectangles, circles, ellipses and arcs using Graphics Class, Color class, ActionListener, ItemListener, KeyListener and MouseListener Interfaces, Component Class, Container Class, Button, Label, Checkbox, Radio Buttons, List Box, Choice Box, Text Area, Border Layout and Grid Layout

Swing

Introduction to Swing, Difference between AWT and Swing, Basics of JOptionPane, JApplet,JButton, JFrame, under javax.swing package.

Text Book and References:

1. E. Balaguruswamy, Programming with Java, 5th Edition, McGraw Hill, 2014.
2. Herbert Schildt, JAVA The Complete Reference, 8th Edition, Mcgraw Higher Education, 2011
3. Cay S. Horstmann, Gary Corness, Core Java Volume II - Advanced Features, 9th Edition, PearsonEducation, 2006.
4. Ken Arnold, James Gosling, David Homes, The Java Programming Language, Addison-WesleyProfessional, 4th Edition, 2005.

MCA-CC106 Programming and Data Structure Cr 3

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

MCA-CC106.CO1.	Design correct programs to solve problems.
MCA-CC106.CO2.	Compare various programming, and apply the concept of decision structures, loops and functions.
MCA-CC106.CO3	Choose efficient data structures and apply them to solve problems.
MCA-CC106.CO4	Analyze the efficiency of programs based on time complexity.
MCA-CC106.CO5	Prove the correctness of a program using loop invariants, pre-conditions and post-conditions in programs

Unit – I
C Language Fundamentals: Character set, Identifiers, keywords, data types, Constants and variables, statements, expression, operators, precedence of operators, Input-output, control statements, control structures.

Unit – II
Arrays, functions, strings, recursion, Tower of Hanoi problem, storage classes, structures, Union, pointer and File Handling.

Unit - III
Stacks and Queues: Notations and Analysis. Storage representation of array, sparse matrix, Insertion and Deletion from an array, merging of two sorted arrays, Stacks and Queues, Application of stack: Infix to Postfix expression, Evaluation of Postfix expression.

Unit- IV
Linked Lists: Singly linked lists, linked stacks and queues, Operations on Polynomials, Linked Dictionary, Doubly Linked Lists, Circular Linked Lists. Dynamic Storage Management.
Binary Tress: Binary search Tree, General Trees, Tree Traversing, Operations on Binary Trees, Expression Tree, Height Balanced Trees.
Graphs: Representation of Graphs - BFS, DFS, Hash Table Methods.

Unit – V
Development of Algorithms: Sorting and Searching Techniques: Selection sort, Bubble sort, Heap sort, Quick sort, linear search, Binary Search.

Text Book and References:

1. Reema Thareja, Data Structures Using C, 2nd edition, Oxford Publication, 2014.
2. Ellis Horowitz, Sartaj Sahani, Fundamentals of Data Structures in C, 2nd edition, University Press, 2008.

MCA-CC107 Discrete Mathematics Cr 3

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

MCA-CC107.CO1.	Explain the various fundamental concepts of the Set theory and Logics.
MCA-CC107.CO2.	Apply the Inclusion and Exclusion Principal to optimize the internal and external validity of the study.
MCA-CC107.CO3.	Derive the solution using deductive logic and prove the solution based on logical inference
MCA-CC107.CO4.	Analyze the concept of Algebraic Structures and evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
MCA-CC107.CO5.	Develop the given problem as graph networks and solve with techniques of graph theory.

Unit I:

Sets, Relation and Function: Operations and Types of Sets, Cartesian Products, Functions and its different types, Inverse and Composite function, Binary Relation and its types, Representation of relations, Partial Ordering Relation, Equivalence Relation, Finite and infinite Sets, Venn diagrams

Principles of Mathematical Induction: The Well-Ordering Principle, Recursive definition, The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The Fundamental Theorem of Arithmetic.

Unit II:

Basic counting techniques-inclusion and exclusion, pigeon-hole principle, permutation and Combination, Recurrences relations

Unit III:

Propositional Logic: Propositions and logical operations, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference.

Proof Techniques: Direct Proof, Indirect Proof- Proof by Contradiction, Proof by Contraposition.

Unit IV:

Algebraic Structures and Morphism: Definition, Properties, Types: Semi Groups, Monoid, Groups, Abelian group, Subgroup, cyclic groups, Factor group, Permutation groups, Normal subgroup, Cosets and Lagrange's Theorem, Homomorphism and Isomorphism of Groups.

Boolean Algebra: Boolean Functions, Representing Boolean Functions, Principal of Duality, Design and Implementation of Digital Networks, Karnaugh maps.

Unit V:

Graphs and Trees: Graph Theory- Basic Concept of Graph Theory and Terminology, representation of Graphs, Bipartite, Regular, Planar and connected graphs. Matrix representation of graphs, Storage representation and manipulation of graphs, Euler graphs, Hamiltonian path and circuits, graph traversals, shortest path in weighted graphs, Graph Isomorphism.

Trees: Introduction, Undirected Trees, Binary search trees, Spanning trees, Minimum spanning trees, Kruskal's Algorithm, Prim's Algorithm.

Suggested Text Books

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill, 7th Edition, 2017.
2. C L Liu and D P Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, Tata McGraw – Hill, 4th Edition, 2017.
3. Susanna S. Epp, Discrete Mathematics with Applications, Wadsworth Publishing Co. Inc., 4th edition, 2011.
4. T.Veerarajan," Discrete Mathematics ". Tata McGraw Hill,2012.

MCA-CC108 Data Analytics Cr 3

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

MCA-CC108.CO1.	Analyze and perform data analysis and statistical analysis to interpret and draw meaningful conclusions from datasets.
MCA-CC108.CO2.	Create and generate effective data visualizations to represent and communicate information derived from the analysis.
MCA-CC108.CO3.	Evaluate and assess and apply critical thinking skills to make informed decisions based on data and deep analytics.
MCA-CC108.CO4.	Apply and utilize technical skills in predictive and prescriptive modeling to support business decision-making processes.
MCA-CC108.CO5.	Synthesize and integrate data analysis results to translate complex data into clear, actionable insights for decision-makers.

Unit I. Basics of Mathematics and Statistics

Introduction to the course and Descriptive Statistics, Introduction to Probability and Probability Distributions, Inferential Statistics through Hypothesis Testing

Unit II. Machine Learning: Introduction and Concepts

Overview of Machine Learning, Regression: Linear Regression, Logistic Regression, KNN (K-Nearest Neighbors)

Unit III. Basic Machine Learning Tools

Decision Trees and Ensemble Learning: Random Forests, Introduction to Clustering: K-Means, Introduction to Neural Networks

Unit IV. Intermediate Machine Learning Techniques

Advanced Classification Techniques: Support Vector Machines, Introduction to Dimensionality Reduction Techniques, Introduction to Naïve Bayes and Model Evaluation Metrics

Unit V. Advanced Topics in Data Analytics

Time Series Analysis and Forecasting, Natural Language Processing (NLP) Basics, Introduction to Recommender Systems

Text Book and References

1. An Introduction to Statistical Learning: with Applications in R, James, Springer
2. Pattern Recognition and Machine Learning, Bishop, Springer
3. Machine Learning (in Python and R) For Dummies, John Paul Mueller, Wiley

MCA-CP102 Programming in JAVA Lab Cr 2

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

MCA-CP102.CO1	Recall fundamental Java programming syntax and tools required for setting up and executing Java programs.
MCA-CP102.CO2	Demonstrate the use of Object-Oriented Programming (OOP) concepts such as inheritance, polymorphism, and abstraction in a lab environment.
MCA-CP102.CO3	Implement Java programs to perform tasks such as data processing, file handling, and basic algorithmic problem-solving.
MCA-CP102.CO4	Analyze the efficiency of Java code by examining execution flow, identifying potential errors, and optimizing logic during debugging.
MCA-CP102.CO5	Develop interactive and functional Java applications integrating graphical user interfaces (GUIs), multithreading, and database connectivity.

List of Experiments

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(, setLength(), append(), insert(), concat()and equals().
9. Write a program to create a ―distance‖ class with methods where distance is computed in termsof feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given in a different file

belonging to the same package.

17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program ―DivideByZero‖ that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronizationamong threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the openConnection()method and then use it examine the different components of the URLand content.
24. Write a program to implement a simple datagram client and server in which a message that istyped into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in thebanner from left to right across the applet‗s window.
26. Write a program to get the URL/location of code (i.e. java code) and document (i.e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using main() function.
30. Write a program to demonstrate the use of push buttons.

MCA-CP104 Programming and Data Structures Lab Cr 2

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

MCA-CP104.CO1.	Demonstrate familiarity with major programming concepts and data structures.
MCA-CP104.CO2.	Calculate and analyze performance of algorithms.
MCA-CP104.CO3.	Choose the appropriate data structure and algorithm design method for a specified application.
MCA-CP104.CO4.	Identify which algorithm or data structure to use in different scenarios.
MCA-CP104.CO5.	Develop algorithms using recursive methods.

List of Experiments

1. Write a C Program to find Factorial of a given number.
2. Write a C Program to find GCD of given two numbers.
3. Write a C Program to solve Towers of Hanoi Problem.
4. Write a C Program to find Factorial of a given number using Recursion.
5. Write a C Program to find GCD of given two numbers using Recursion.
6. Write a C Program to solve Towers of Hanoi Problem using Recursion.
7. Write a C program to find both the largest and smallest number in a list of integers.
8. Write a C program to perform addition of two matrices using function.
9. Write a C program to perform multiplication of two matrices using function.
10. Write a C program to insert a sub-string in to a given main string from a given position using string function.
11. Write a C program to delete n characters from a given position in a given string using string function.
12. Write a C program that displays the position or index in the string S where the string T begins, or –1 if S doesn‘t contain T.
13. Write a C program to count the lines, words and characters in a given text.
14. Write a C Program to perform various arithmetic operations on pointer variables.
15. Write a C Program to demonstrate Call by Value
16. Write a C Program to demonstrate Call by Reference
17. Write a C program that uses functions to perform following operations on complex numbers
i) Reading ii) writing iii) addition iv) multiplication
18. Write C programs that implement stack (its operations) using arrays.
19. Write C programs that implement stack (its operations) using pointers.
20. Write C programs that implement Queue (its operations) using arrays.
21. Write C programs that implement Queue (its operations) using pointers.
22. Write a C program that uses Stack operations to perform converting infix expression into postfix expression.
23. Write a C program that uses Stack operations to perform evaluating the postfix expression.
24. Write a C program that uses functions to perform the creation, insertion, deletion and traversal operations on singly linked list.

25. Write a C program that uses functions to perform the creation, insertion, deletion and traversal operations on doubly linked list.

26. Write a C program that uses functions to perform the creation, insertion, deletion and traversal operations on circular linked list.

27. Write a C program that uses functions to perform the following

i)Create a Binary Tree of integers

ii)Traverse the above binary tree in preorder, inorder and postorder

28. Write a C program to perform linear search operation for a key value in a given list of integers.

29. Write a C program to perform binary search operation for a key value in a given list of integers.

30. Write a C program to perform linear search operation for a key value in a given list of integers using recursive function.

31. Write a C program to perform binary search operation for a key value in a given list of integers using recursive function.

32. Write a C program that implements bubble sort to sort given list of integers in ascending order.

33. Write a C program that implements selection sort to sort given list of integers in ascending order.

34. Write a C program that implements insertion sort to sort given list of integers in ascending order.

MCA-CP105 Data Analytics Lab Cr 2

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

MCA-CP105.CO1	Execute and conduct data analysis and statistical procedures proficiently within a laboratory setting.
MCA-CP105.CO2	Produce and create effective data visualizations to illustrate findings and insights derived from experimental data.
MCA-CP105.CO3	Evaluate and employ critical thinking skills to assess experimental results and make informed decisions based on data analysis within the lab context.
MCA-CP105.CO4	Apply and utilize technical skills in predictive and prescriptive modeling to support experimental design and decision-making processes in the laboratory.
MCA-CP105.CO5	Synthesize and integrate laboratory data analysis outcomes to generate clear and actionable recommendations for experimental optimization and future research directions.

List of Experiments

Program 1: Matrix Multiplication Using Numpy Array.
Program 2: Sorting Python Dictionary.
Program 3: Compute Interquartile Range Ofa Given Datasample.
Program 4: Shapiro-Wilk Test To Verify Gaussian Distribution.
Program 5: Computing Pearson’s Correlation Coefficient.
Program 6: Conducting Chi-Squared Test.
Program 7: Program To Implement Student’s t-Test.
Program 8: ANOVA.Test.
Program 9: Implementing Fancy Indexing In Python.
Program 10: Feature Scaling Using Min-Max Normalization.
Program 11: Z-Score Standardization.
Program 12: Data Visualization Using Python.
Program 13: Finding Feature Importance Using Linear Regression.
Program 14: Classification Using Decision Tree.
Program 15: Classification Using KNN.
Program 16: Implementing PCA.
Program 17: Classification Using Random Forest

SEMESTER - II

MCA-CC201 Software Engineering Cr 4

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

MCA-CC201.CO1	Define the fundamental concepts, components, and processes of software engineering, including software quality attributes and process models.
MCA-CC201.CO2	Explain the principles of requirement engineering, requirement elicitation techniques, and the use of modeling tools such as DFDs and ER diagrams for software requirement documentation.
MCA-CC201.CO3	Apply software design principles, including modularity, coupling, cohesion, and object-oriented design, to create structured software solutions.
MCA-CC201.CO4	Analyze software testing strategies and methods, such as white-box and black-box testing, to ensure functionality, performance, and reliability of software systems.
MCA-CC201.CO5	Develop a comprehensive plan for software maintenance and reengineering, considering the categories and costs of maintenance and the need for software evolution.

Unit I. Introduction to SE
Definitions and components of SE, Software processes and S/W characteristics, S/W applications. Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes, Software Process Models: Waterfall, Spiral, Prototype, RAD, 4 GT, Incremental, Evolutionary development, Iterative enhancement, Selection of a model.
Software Metrics
Definitions, Types of S/W metrics, products and processes S/W size and effort estimations: LOC, KLOC etc. Function point estimations COCOMO: basic, intermediate, detailed modes for organic, semidetached and embedded S/W.

Unit II. Software Requirements Analysis and Specification
Requirement engineering, requirement elicitation, DFD, ER diagrams, Charts, requirement analysis,requirement documentations.

Unit III. Software Design
Definitions, concepts and principles of S/W design, Structured design: modularity, coupling, cohesion, OO design principles.

Unit IV. S/W Testing
Objective, principles, types of S/W testing- Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Alpha and Beta Testing of products, S/w quality assurance: concepts of S/W quality, S/W reliability, S/W safety.

Unit V: Software Maintenance

Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software ReEngineering, Reverse Engineering.

Text Book and References

1. Rogers G. Pressman, Software Engineering: A Practitioner‘s Approach (7th Edition),McGraw-Hill, 2009.
2. Rajib Mall, Fundamentals of Software Engineering, PHI, 2014.
3. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.

MCA-CC203 Database Management System Cr 3

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

MCA-CC203.CO1	Define key concepts of transaction processing, including properties of transactions, concurrency control mechanisms, and types of database recovery techniques.
MCA-CC203.CO2	Explain database security goals, threats, and mechanisms such as authorization, authentication, discretionary access control, and audit trails.
MCA-CC203.CO3	Implement object-oriented database principles and PL/SQL constructs such as cursors, triggers, procedures, and functions to solve database-related problems.
MCA-CC203.CO4	Analyze the architecture and advantages of parallel databases, decision support systems (DSS), and data warehouses, identifying their limitations and applications.
MCA-CC203.CO5	Design and develop decision support systems and implement data mining processes to extract actionable insights from large datasets.

Unit I. Transaction Processing Concepts

Transaction, Transaction execution, transaction processing and transaction properties, Concurrency control: Concurrency control, problems of concurrency control i.e., Lost update, dirty read (uncommitted data), unrepeatable (read), schedules, Lock types i.e. Shared, exclusive, two phase (2PL)), dead-locks, timestamp, methods.

Unit II. Database Security (DS)

Goals of DS, Threats to DS, Authorization and authentication, data security issues, discretionary Accesscontrol (i.e. granting / revoking privileges, Audit trails).

Database recovery system

Database Backup, and Types of DB failures, types of Database recovery i.e. Forward and backward, recovery, Recovery technique i.e. Deferred update, immediate update and checkpoints.

Unit III. Object oriented databases

Object oriented data model, concepts of object oriented databases (i.e. Objects, classes, polymorphism), Object Oriented DBMS i.e. OODBMS (Features, advantages and disadvantages).

Parallel Database System: Advantages, disadvantages and architecture of Parallel DBS.

Unit IV. PL/SQL

Basics of PL/SQL, Cursors, Triggers, Procedures and Functions.

Unit V. Decision Support System (DSS):

Evolution of DSS, Definition of DSS,Characteristics of DSS, Benefits of DSS, Components of DSS.

Data Warehousing

Evolution, Components, Characteristics, Benefits, Limitation, Architecture of Data Warehouses

(Data marts and Online Analytical Processing(OLAP)).

Data Mining

Data Mining Process, Data Mining Knowledge discovery, Goals of Data Mining, Data Mining Tools, Data Mining Application.

Text Books

1. Database System Concepts by Silberschatz, Korth & Sudarshan (McGraw-Hill Education)
2. Fundamentals of Database System By Elmasari & Navathe- Pearson Education

Reference Books

1. Database Management Systems by RamaKrishna & Gehrke (McGraw-Hill Education)
2. Fundamentals of Relational Database management Systems by Sumathi & Esakkirajan, Springer

MCA-CC204 Operation Research and Optimization Techniques Cr 4

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

MCA-CC204.CO1	Define the fundamental concepts, models, and significance of Operations Research (OR) in quantitative decision-making and its applications in management.
MCA-CC204.CO2	Explain the principles and methods of linear programming, including the simplex method, graphical methods, and degeneracy handling in optimization problems.
MCA-CC204.CO3	Apply duality concepts in linear programming and solve primal and dual problems using dual simplex methods and post-optimal analysis techniques.
MCA-CC204.CO4	Analyze and solve integer programming problems using Gomory's fractional cut method, revised simplex techniques, and parametric linear programming.
MCA-CC204.CO5	Formulate and solve nonlinear programming problems (NLPP) by applying Kuhn-Tucker conditions and constrained optimization methods for complex decision-making

Unit 1. Introduction
Decision Making, Quantitative Approach to Decision Making, Nature and Significance of OR in Decision Making, Scientific Methods in Operations Research, Models in Operations Research, Application Areas of OR in Management.

Unit 2. Linear Programming
Model Formulation, Graphical Methods, Simplex Method, Big M Method, Two Phase Technique, Maximization and Minimization of L.P.P, Degeneracy in L.P.P.

Unit 3. Duality in LPP
Introduction, General Primal – Dual pair, Formulating a Dual Problem, Primal Dual pair in Matrix form, Dual simplex method, Post optical Analysis.

Unit 4. Integer Programming and Advance LPP techniques:
Introduction, Gomory’s Method, Courtruction of Gomory’s constraints. Fractional Cut Method: All Integer & Mixed Integer, Revised Simplex Method, Bounded Variable, Parametric LPP.

Unit 5. NLPP
Introduction, Formation a NLPP, General NLPP, Constrained Optimization with Equality Constraints and Inequality constraints, Saddle Points. Graphical solution, Kuhn – Tucker Conditions with Non-Negative Constraints.

Text Books & Reference Books:

1. K. Swarup, P.K. Gupta, and M. Mohan, Operations Research, S. Chand, 2010.
1. H. M. Taha, Operations Research, Pearson, 10th Edition, 2019.
2. J.K. Sharma, Operations Research: Theory and Application, MACIN, 2012.

MCA-CC206 Artificial Intelligence & Machine Learning Cr 3

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

MCA-CC206.CO1.	Understand the modern view of AI as the study of agents that receive percepts from the environment and perform actions.
MCA-CC206.CO2.	Demonstrate awareness of the major challenges facing AI and the complex of typical problems within the field.
MCA-CC206.CO3.	Exhibit strong familiarity with a number of important AI techniques, including in particular search, knowledge representation, planning and constraint management.
MCA-CC206.CO4.	Asses critically the techniques presented and to apply them to real world problems.
MCA-CC206.CO5.	Develop the ability to integrate AI and machine learning techniques to design, implement, and evaluate systems for solving complex, real-world problems while considering ethical and societal implications.

Unit 1: Introduction to Artificial Intelligence and Search Strategies

This unit introduces Artificial Intelligence, its applications, and the formulation of AI problems. It covers Intelligent Agents, their types, environments, PEAS representation, and the architecture of intelligent agents. The unit also explores reasoning and logic, including propositional logic, first-order logic, inference methods, forward and backward chaining. The second part focuses on search strategies, addressing problem-solving through search, issues in search program design, and various search techniques such as uninformed search (BFS, DFS), heuristic search methods (Generate-and-Test, Hill Climbing, Best-First Search, A* Algorithm, Alpha-beta search algorithm), and advanced techniques like Problem Reduction, AO* Algorithm, Constraint Satisfaction, and Means-Ends Analysis.

Unit 2: Machine Learning Fundamentals

This unit covers the basics of machine learning, its applications, and its distinction from data mining and big data analytics. Supervised learning methods such as Naive Bayes Classifier, k-Nearest Neighbor Classifier, and Decision Tree Classifier are introduced. Unsupervised learning techniques, including K-means Clustering and Association Analysis using the Apriori Algorithm, are explained. The unit also includes an introduction to reinforcement learning.

Unit 3: Artificial Neural Networks and Optimization Techniques

This unit explores artificial neural networks (ANNs), beginning with foundational concepts such as activation functions and optimization techniques like Gradient Descent. It delves into different network architectures, including Perceptrons, Adaline, and Multilayer Perceptrons, with a focus on Backpropagation algorithms, training procedures, and tuning network size. The unit also discusses forecasting and learning theory, covering non-linear regression, logistic regression, random forest, Bayesian belief networks, bias/variance tradeoff, model complexity tuning, and model selection dilemmas. Clustering techniques such as the Expectation-Maximization Algorithm, hierarchical clustering, supervised learning post-clustering, and choosing the number of clusters are also included.

Unit 4: Advanced Machine Learning and Kernel Machines

This unit introduces kernel machines and advanced machine learning techniques. It discusses the concept of optimal separating hyperplanes, separating data with maximum margin, and the Support Vector Machine (SVM) methodology, including soft margin hyperplanes and the kernel trick. Ensemble methods are covered in detail, including mixture models, classifiers using multiple data samples, error-focused improvement techniques, and weak learners with decision stumps. Techniques such as bagging, stacking, boosting, and implementing AdaBoost are explained, along with methods like bootstrapping and cross-validation.

Unit 5: Dimensionality Reduction and Feature Selection

This unit focuses on dimensionality reduction techniques such as subset selection, Principal Components Analysis (PCA), multidimensional scaling, and Linear Discriminant Analysis (LDA). Feature selection is also explored, including feature ranking and subset selection, providing a comprehensive understanding of methods to optimize feature representation and model efficiency.

Text Book and References:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell &Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2ndedition, 2005.
3. Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.
5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education,3rd edition, 2000.

MCA-CC207 Computer Networks Cr 3

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

MCA-CC207.CO1	Remember basic computer network technology
MCA-CC207.CO2	Understand and explain Data Communications System and its components
MCA-CC207.CO3	Analyze the different types of network topologies and protocols
MCA-CC207.CO4	Apply the terminology and concepts of the OSI reference model and the TCP-IP reference model.
MCA-CC207.CO5	Implement the basic protocols of computer networks, and their usage to assist in network design.

Unit I. Basics of Digital Communication

Introduction to digital communication, Signal propagation, Signal types, Signal parameters , Switching & forwarding, Transmission impairments, Attenuation, Delay distortion, Noise, Effects of limited bandwidth, Data rate limits-Nyquist’s theorem and Shannon’s theorem.

Unit II. Network Organization and Models

Basics of computer Network, topology & types of topologies, types of networks(LAN, MAN, WAN), types of communications (Asynchronous and synchronous), modes of communications (simplex, half duplex, full duplex), Protocols, Networking models, ISO-OSI Reference Model, Design issues of the layer, Internet Model (TCP/IP), Comparison of ISO-OSI & TCP/IP Model.

Networking Devices

Connectivity DevicesPassive & Active Hubs, Repeaters, Switches (2-Layer Switch, II Devices 3-Layer switch (Router), Bridges (Transparent Bridges, Spanning Tree, Bridges, Source Routing Bridges), Gateways.

Unit III: Application, Presentation &Session Layer

Principles of Application Layer Protocols, The Web and HTTP, FTP, Telnet, Electronic Mail in the Internet (SMTP, MIME, POP3, IMAP), DNS, Introduction to SNMP.

Unit IV: Transport Layer & Network Layer

Transport-Layer Services, port addressing, Multiplexing and Demultiplexing, Principles of Reliable Data Transfer, Congestion Control, TCP’s Congestion Control mechanisms.

Network Service Model, Data gram & Virtual Circuit, Routing Principles, The Internet Protocol, (ipv4 & ipv6), IP addressing and subnetting, Routing Algorithms.

Unit V: Data Link Layer & Physical Layer

Data Link Layer, Error Detection and Correction Techniques, Multiple Access Protocols, LAN Addresses and ARP & RARP, PPP: The Point-to-Point Protocol, Ethernet standards – IEEE 802.3, 802.5, FDDI, 802.6.

Physical Layer, Types of media wired and wireless media.

Text Books and References

1. Data communication & Networking, Fourozan, THM.
2. Computer Network. Tannenbaum, PHI.
3. Data and Computer Communications, Stallings, PHI.
4. Data Communication and Network, Dr. Prasad, Wiley, Dreamtech.

MCA-CP201 Data Communication and Computer Networks Lab Cr 2

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

MCA-CP201.CO1	Understand the working of various network topologies and circuit and packet switching.
MCA-CP201.CO2	Comprehend the role of data link layers and significance of MAC protocols.
MCA-CP201.CO3	Understand the networking protocols and Internet protocols.
MCA-CP201.CO4	Understand the transport layer working with TCP, UDP and ATM protocols.
MCA-CP201.CO5	Comprehend the functionality of application layer and importance of network security.

List of Experiments

1. To study the basic networking commands.
2. To write a socket program for implementation of echo.
3. To write a client-server application for chat using TCP
4. To Perform File Transfer in Client & Server Using TCP/IP.
5. To implement Remote Command Execution (RCE).
6. To write a program to implement simple client-server application using UDP.
7. To implement Address Resolution Protocol.
8. To download a webpage using Java
9. To write a socket program for implementation of TCP module.
10. To implement Remote Method Invocation.

MCA-CP204 Artificial Intelligence & Machine Learning Lab Cr 2

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

MCA-CP204.CO1.	Execute and conduct data analysis and statistical procedures proficiently within a laboratory setting.
MCA-CP204.CO2.	Produce and create effective data visualizations to illustrate findings and insights derived from experimental data.
MCA-CP204.CO3.	Evaluate and employ critical thinking skills to assess experimental results and make informed decisions based on data analysis within the lab context.
MCA-CP204.CO4.	Apply and utilize technical skills in predictive and prescriptive modeling to support experimental design and decision-making processes in the laboratory.
MCA-CP204.CO5.	Synthesize and integrate laboratory data analysis outcomes to generate clear and actionable recommendations for experimental optimization and future research directions.

List of Experiments

1. Introduction to Java, Python or R to implement the assignments.
2. Introduce and implement different Supervised and Unsupervised learning techniques.
3. Optimization method implementation like Genetic Algorithm.
4. Problem solving by Soft Computing Techniques-Fuzzy Logic.
5. Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets for real life applications.
6. Develop a program to solve a CSP, such as a Sudoku solver or a map-coloring problem.
7. Implement heuristic search techniques such as Hill Climbing and Simulated Annealing for optimization problems.
8. Create a Decision Tree Classifier and evaluate its performance on a dataset.
9. Implement the K-Means clustering algorithm and visualize the clusters formed for a given dataset.
10. Develop a Naive Bayes Classifier and test its performance on a text classification dataset such as spam detection.
11. Write a program to classify a dataset using Support Vector Machines and experiment with different kernel functions.
12. Perform PCA on a high-dimensional dataset to reduce dimensionality and visualize the principal components.
13. Implement the Apriori algorithm for association rule mining and demonstrate its application in market basket analysis.
14. Develop a program to implement an ensemble technique such as Random Forest or Gradient Boosting, and compare its performance with standalone classifiers.
15. Apply Linear Discriminant Analysis (LDA) to a dataset and analyze its impact on classification tasks.

MCA-CP205 Database Management System Lab Cr 2

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

MCA-CP205.CO1.	Design conceptual database models using ER diagrams, identifying entities, attributes, and relationships.
MCA-CP205.CO2.	Convert ER models into normalized relational schemas up to Third Normal Form (3NF).
MCA-CP205.CO3.	Execute commands to create, alter, and manipulate database structures and data.
MCA-CP205.CO4.	Construct advanced queries using subqueries, joins, aggregate functions, and constraints.
MCA-CP205.CO5.	Implement database programming constructs like triggers, stored procedures, and cursors for automation and optimization.

List of Experiments

Experiment 1

Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.

Experiment 2

Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, represent attributes as columns, identifying keys) Note: Student is required to submit a document showing the database tables created from ER Model.

Experiment 3

Normalization -To remove the redundancies and anomalies in the above relational tables, Normalize up to Third Normal Form

Experiment 4

Creation of Tables using SQL- Overview of using SQL tool, Data types in SQL, Creating Tables (along with Primary and Foreign keys), Altering Tables and Dropping Tables

Experiment 5

Practicing DML commands- Insert, Select, Update, Delete

Experiment 6

Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT, CONSTRAINTS etc.

Experiment 7

Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).

Experiment 8

Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.

Experiment 9

Practicing on Triggers - creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger

Experiment 10

Procedures- Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

Experiment 11

Cursors- Declaring Cursor, Opening Cursor, Fetching the data, closing the cursor.

SEMESTER - III

MCA-CC301 Values and Ethics of Profession Cr 3

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

MCA-CC301.CO1	Recall the fundamental concepts of ethics, values, and professional conduct in the context of technology and business environments.
MCA-CC301.CO2	Explain the importance of ethical decision-making and the role of professional values in the technology and business sectors.
MCA-CC301.CO3	Apply ethical principles and professional values to real-world scenarios in the workplace, particularly in computing and technology industries.
MCA-CC301.CO4	Analyze ethical dilemmas and challenges that professionals face in the workplace and evaluate potential solutions using ethical frameworks.
MCA-CC301.CO5	Design and implement a personal code of ethics, demonstrating an understanding of ethical responsibilities and professional conduct in various situations.

Unit 1: Introduction to Professional Ethics

Overview of Ethics and Values: Definitions, importance, and scope of ethics in professional settings.
Ethical Theories: Deontology, Utilitarianism, Virtue Ethics, and Relativism.
The Role of Ethics in Technology and Business: Ethical decision-making in the context of technology-driven industries.
Ethical Principles: Honesty, Integrity, Accountability, and Transparency.

Unit 2: Code of Ethics and Professional Responsibility

Global Codes of Ethics: Understanding professional codes like ACM, IEEE, and other industry-specific codes.
Professional Conduct: The responsibilities of professionals toward clients, colleagues, society, and the environment.
Confidentiality and Data Protection: Ethical responsibilities related to privacy, data security, and intellectual property.
Social and Environmental Responsibility: Addressing societal impacts of technological developments.

Unit 3: Ethical Decision Making in Technology and Business

Ethical Dilemmas in Technology: Privacy issues, cybersecurity, AI ethics, and the digital divide.
Corporate Social Responsibility (CSR): Business ethics, corporate governance, and sustainability.
Case Studies: Analysis of real-world ethical dilemmas in the tech industry (e.g., data breaches, misuse of AI, etc.).
Making Ethical Decisions: Tools and frameworks for ethical decision-making (e.g., the ethical decision-making model, stakeholder analysis).

Unit 4: Ethical Leadership and Professionalism

Leadership and Ethics: Role of ethical leadership in creating ethical workplace cultures.
Leadership Styles: Transformational leadership, servant leadership, and ethical decision-making.
Managing Ethical Conflicts: Techniques for resolving conflicts based on ethical principles and professional values.
Promoting Ethical Behavior: Encouraging ethical conduct among employees and colleagues.

Unit 5: Ethics in the Workplace and Society

Workplace Ethics: Professional relationships, respect, diversity, and inclusivity in the workplace.
Whistleblowing and Ethical Challenges: Legal and moral aspects of whistleblowing.
Ethics of Artificial Intelligence and Automation: Exploring ethical implications in AI, automation, and their impact on jobs and society.
Global Ethical Issues: Addressing ethical challenges in a globalized world, such as outsourcing, environmental impact, and labor practices.

Text Book and References:

1. G. Reynolds, “Ethics in Information Technology”, Cengage Learning, 2019
2. Richard A. Spinello, “Computer Ethics: A Case-based Approach" Pearson Education, 2017
3. George W. Ford, William M. Oliver, and Judith D. Rainer, “Professional Ethics in Information Technology”, Course Technology, 2006

MCA-CC302 Accountancy and Financial Management Cr 3

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

MCA-CC302.CO1	Explain & examine the financial Accounting concept and its important with generally accepted accounting principles ,Balance sheet and its inventory valuation with Depreciation value
MCA-CC302.CO2	Demonstrate & Examine the Fund flow statement and cash flow statement for management decision
MCA-CC302.CO3	Analyze & Implement cost accounting process, its control methods and can examine the reconciliation between Cost accounting and financial accounting
MCA-CC302.CO4	Evaluate and Examine overhead cost control and budgetary control and record basic accounting transactions and prepare annual financial statements to improve organizational culture and climate.
MCA-CC302.CO5	Critically analyze, interpret and communicate the information contained in basic financial statements and explain the limitations of such statements.

Unit I. Overview: Accounting concepts, conventions and principles; Accounting Equation, International Accounting principles and standards**;** Matching of Indian Accounting Standards with International Accounting Standards.

Unit II. Mechanics of Accounting: Double entry system of accounting, journalizing of transactions; preparation of final accounts, Trading Account, Manufacturing Accounts,Profit & Loss Account, Profit & Loss Appropriation account and Balance Sheet, Policies related with depreciation, inventory and intangible assets like copyright, trademark, patents and goodwill.

Unit III. Analysis of financial statement: Ratio Analysis- solvency ratios, profitability ratios, activity ratios, liquidity ratios, market capitalization ratios ; Common Size Statement ; Comparative Balance Sheet and Trend Analysis of manufacturing, service & banking organizations.

Unit IV. Funds Flow Statement: Meaning, Concept of Gross and Net Working Capital, Preparation of Schedule of Changes in Working Capital, Preparation of Funds Flow Statement and its analysis ; Cash Flow Statement: Various cash and non-cash transactions, flow of cash, preparation of Cash Flow Statement and its analysis.

Unit V. Financial Decisions: Management of Working Capital, Managing Cash Needs, Capital Structure,Dividend Decisions. Introduction to financial accounting computer packages (TALLY).

Reference Books:

1. “Book Keeping and Accountancy” Choudhari, Chopde.
2. “Cost Accounting”: Choudhari, Chopde.
3. “Financial Management” Text and Problems: M.Y.Khan, P.K. Jain.
4. “Financial Management Theory & Practice” Prasanna Chandra Tata McGraw Hill.
5. Managerial Economics & Financial Analysis, Siddiqui S.A. Siddiqui A.S. New Age.

MCA-CP302 Accountancy Lab Cr 2

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

MCA-CP302.CO1	Understand the fundamentals of accounting principles and concepts within the context of Tally software.
MCA-CP302.CO2	Develop proficiency in navigating and using Tally software for various accounting tasks, including ledger creation, journal entries, and financial statement generation.
MCA-CP302.CO3	Analyze financial data generated by Tally to interpret and evaluate the financial performance of businesses, enabling informed decision-making.
MCA-CP302.CO4	Apply advanced features of Tally software such as inventory management, taxation, and payroll processing to simulate real-world accounting scenarios.
MCA-CP302.CO5	Troubleshoot common issues and errors encountered while using Tally software, employing problem-solving techniques to ensure smooth accounting operations.

List of Experiments

Software package used for Accountancy: Tally accounting package

1. Accounts Creation
2. Inventory Creation
3. Transactions

a. Accounting vouchers

b. Inventory vouchers

4. Importing of data
5. Reports
6. Introduction to Payroll system

MCA-CP303 Seminar and Technical Writing Cr 2

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

MCA-CP303.CO1.	Recall key principles of technical writing, including structure, clarity, and presentation techniques for technical documents and presentations.
MCA-CP303.CO2.	Understand the various types of technical documents (e.g., reports, research papers, proposals) and their specific purposes in professional settings.
MCA-CP303.CO3.	Apply technical writing skills to prepare well-structured research papers, reports, and project documentation, adhering to academic and industry standards.
MCA-CP303.CO4.	Analyze and critique technical documents, identifying strengths and areas for improvement in clarity, organization, and formatting.
MCA-CP303.CO5.	Design and deliver an effective seminar presentation on a technical topic, integrating research, data analysis, and professional communication skills.

Unit I. Introduction to Technical Writing

Overview of Technical Writing, Types of Technical Documents: Reports, Proposals, Research Papers, User Manuals, etc. Structure and Organization of Technical Documents, Importance of Clarity, Conciseness, and Readability in Technical Writing, Common Technical Writing Mistakes and How to Avoid Them

Unit II. Writing Techniques for Technical Documents

Writing an Abstract, Introduction, and Conclusion, Creating Effective Figures, Tables, and Graphs Using Citations and References in Technical Writing (APA, IEEE, etc.), Writing Project Reports, Research Papers, and Proposals, Preparing Executive Summaries for Technical Reports

Unit III. Research and Data Analysis

Conducting Literature Reviews and Research for Technical Writing, Structuring and Presenting Data: Statistical Analysis, Graphs, Charts, Ethical Issues in Technical Writing: Plagiarism and Citation Ethics, Creating Well-Structured Sections: Methodology, Results, Discussion, Writing for a Specific Audience: Technical and Non-Technical Readers.

Unit IV. Preparing and Delivering a Technical Seminar

Preparing a Seminar: Topic Selection, Research, and Content Organization, Writing a Seminar Paper: Abstract, Introduction, Body, Conclusion, Oral Presentation Skills: Structure, Visual Aids, and Public Speaking, Engaging the Audience: Techniques for Effective Delivery and Q&A Sessions, Evaluating the Success of a Seminar: Feedback and Reflection

Unit V. Professional Communication in the Industry

Writing Emails, Memos, and Business Correspondence, Writing Proposals and Documentation for Software Projects, Creating Technical Documentation for IT Products (e.g., user manuals, software documentation), Importance of Communication in Collaborative Technical Projects, Writing for Global Audiences: Cultural Sensitivity and Clear Communication

i. Every MCA students are required to undertake Seminar and Technical Writing course which is a part of the MCA III semester examination, equivalent to 2 credits carrying 100 marks.

ii. This course provides a comprehensive overview of writing skills crucial for their academic and professional success. It equips students with skills to create clear, concise and accurate technical documents and at the same time focuses on developing presentation and communication skills.

iii. Students can choose topics relevant to computer science and any cutting-edge technologies on their own or by taking help from any faculty member of the department.

iv. Each student has to prepare a research paper/ technical report and needs to communicate and/or publish in any conference/ journals/ book chapters or any such form.

v. The evaluation is done during End-semester, where a student shall be required to make a presentation of his/her technical paper/report. Out of 100 marks, 40 marks will be evaluated based on topic chosen and technical report, 20 marks for Presentation, 20 marks for Viva and 20 marks for publication.

SEMESTER - IV

MCA-CP402 Major Project and Dissertation Cr 12

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

MCA-CP402.CO1.	Recall the project development lifecycle, including requirements gathering, design, implementation, testing, and documentation, essential for completing a major project.
MCA-CP402.CO2.	Understand how to apply theoretical knowledge and practical skills to solve real-world problems by developing a software or IT solution in the form of a project.
MCA-CP402.CO3.	Apply project management techniques, such as task planning, resource allocation, and time management, to effectively execute and deliver the major project.
MCA-CP402.CO4.	Analyze complex problems, design suitable software solutions, and evaluate their effectiveness based on project requirements, user feedback, and testing results.
MCA-CP402.CO5.	Create a comprehensive dissertation documenting the research, development process, methodologies, results, and conclusions, demonstrating critical thinking, innovation, and professional writing skills.

In major project the student has to go for internship in reputed organizations/ institutes for a period of 3-4 months. The objective of the internship is to provide students an insight regarding internal working of companies. With an aim to go beyond academics, internship provides student a practical perspective on the world of work.

It provides students with an opportunity to learn practically through interaction, working methods and employment practices. It gives them exposure to current work practices, projects going on in the organization. Internship provide an excellent opportunity to carry out projects as per recent trends and requirements.

i. Major Project and Viva-voce is an integral part of MCA programme. MCA students are required to undertake Major Project for a period of one semester in a reputed organization connected with industry, consultancy, trade or commerce approved by the Head of the Faculty of Computing and Information Technology. Major Project is a part of the MCA VI Semester examination, equivalent to 12 credits carrying 100 marks.

ii. The topic/title of the Major Project and the name of the supervisor earmarked shall be approved by Project Guide from the industry or organization and/or by an Internal Project Guide from the University. Every MCA student shall also be required to prepare a Project Report/ Dissertation under the supervision of a Project Guide.

iii. Final Project Report shall be submitted to the Head of the Department duly signed by the Project Guide, minimum in triplicate on or before a specified date fixed for this purpose of evaluation by the Head of the Department / Dean of the Faculty of Computing and Information Technology. Students are expected to maintain academic integrity of the submitted project report.

iv. During End-semester evaluation, every student shall be required to make a presentation of his/her project report, including execution of Project work undertaken by him/her during the semester before the duly constituted committee formed by the Head/Dean of the department.

v. Each student may be asked to present Synopsis of the Project Study before the members of the Faculty for consideration and finalization of Project Study Title.

vi. Once the project report is submitted to and approved by the project guide, the same shall be finally examined by an external examiner and internal examiner(s). The external examiner shall evaluate it independently out of 50 marks. The internal examiner(s) shall evaluate it out of 50 marks (average of the marks will be counted if more than one internal examiner remains present). Both external and internal examiners shall evaluate on the basis of project report which carries 20 marks, presentation of the project work which carries 20 marks and viva – voce which carries 10 marks.

MCA-CP403 **Comprehensive Viva** **Cr 2**

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

MCA-CP403.CO1.	Recall key concepts, theories, and techniques learned throughout the MCA program, across various subjects including programming, database management, networking, and software engineering.
MCA-CP403.CO2.	Demonstrate an understanding of the interconnections between different subjects and how they apply to real-world IT scenarios.
MCA-CP403.CO3.	Apply theoretical knowledge and practical skills in a simulated professional environment during the viva to address complex IT challenges and questions.
MCA-CP403.CO4.	Analyze and discuss various technical problems and case studies, providing insights and solutions based on the core principles of computer science and IT.
MCA-CP403.CO5.	Critically evaluate one’s own academic work and the work of peers, responding to questions and providing thoughtful, clear, and concise explanations in the viva format.

i. The comprehensive viva voce examination should be based on the theoretical knowledge, skills and the practices which the students have undergone in the period of two years. It is based on all the courses the students have studied, the political, social and economic developments in the country and around the world.

ii. Evaluation is done during End semester by external examiner as well as internal faculty members. The external examiner shall evaluate it independently out of 50 marks. The internal examiner(s) shall evaluate it out of 50 marks (average of the marks will be counted if more than one internal examiner remains present).

ELECTIVE COURSES (SET A)

MCA-ECA301 Cryptography Cr 4

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

MCA-ECA301.CO1	Understand the basic concepts of security principles, various classical cryptosystems and be familiar with the common threats faced today.
MCA-ECA301.CO2	Design classical encryption techniques and block ciphers
MCA-ECA301.CO3	Design hash and MAC algorithms, and apply the concept of digital signatures for solving problems.
MCA-ECA301.CO4	Evaluate a range of access control and authentication mechanisms.
MCA-ECA301.CO5	Examine and classify different malicious programs, different network issues and the solutions for them through firewall, intrusion detection system and Electronic Mail Security

Unit I. Introduction

Basic Objectives of Cryptography, Secret-key and Public-key cryptography, Security Goals and Principles, Cryptographic Attacks, Substitution Ciphers, Transpositions, Stream and Block Ciphers, Algorithm Modes.

Unit II. Symmetric Key Cryptography

Modern Block Ciphers, Modern Stream Ciphers, Modular Arithmetic, Linear Congruence, GF(2^n) Fields, Diffie-Hellman Key Exchange Algorithm, Data Encryption Standard (DES), Blowfish, Advanced Encryption Standard (AES).

Unit III. Asymmetric Key Cryptography

Overview, Primes, Primality Testing, Factorization, Chinese Remainder Theorem, RSA, Cryptographic Hash function: MD5, SHA, MAC, HMAC, Digital Envelope,Digital Signature.

Unit IV. Entity Authentication and Key Management

Passwords, Challenge-Response, Zero-Knowledge, Kerberos, PKI.

Unit V. Network Security

Threats in Network, Network Security Controls, Firewalls, Intrusion Detection Systems, SecureE-mail, Malicious Programs.

Text Book and References:

1. Cryptography and Network Security: Second Edition, Behrouz A. Forouzan,McGraw Hill Education
2. Network Security Essentials : Applications and Standards: Fourth Edition, William Stalling, Pearson Education
3. Cryptography and Network Security: Atul Kahate, 2nd Edition, Tata McGraw-Hill
4. Applied Cryptography: Bruce Schneier, John Wiley & Sons
5. Security in Computing: P. Pfleeger, , PHI

MCA-ECA302 Compiler Design Cr 4

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

MCA-ECA302.CO1	Describe the design of a compiler and the phases of program translation from source code to executable code and the files produced by these phases.
MCA-ECA302.CO2	Explain lexical analysis phase and its underlying formal models such as finite state automata, push-down automata and their connection to language definition through regular expressions and grammars.
MCA-ECA302.CO3	Explain the syntax analysis phase and identify the similarities and differences among various parsing techniques and grammar transformation techniques.
MCA-ECA302.CO4	Identify the effectiveness of optimization and explain the differences between machine-dependent and machine-independent translation.
MCA-ECA302.CO5	Write a scanner, parser, and semantic analyzer without the aid of automatic generators

Unit I. Overview of Compilation:
Introduction to Compiler, Phases of Compilation, Grouping of Phases.

Unit II. Lexical Analysis:
Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Finite state machines and regular expressions and their applications to lexical analysis.

Unit III. Syntax Analysis:
Context-free grammars, Top-down Parsing – Backtracking, LL(1), recursive descent parsing, Predictive parsing, Bottom-up parsing – Shift Reduce parsing, LR and LALR parsing, Error recovery in parsing, handling ambiguous grammar.

Unit IV. Semantic analysis:
Intermediate forms of source Programs – abstract syntax tree, polish notation and three address codes. Attributed grammars, Syntax directed translation, Conversion of popular Programming languages language Constructs into Intermediate code forms, Type checker.

Symbol Tables:
Symbol table format, organization for block structures languages, hashing, tree structures representation of scope information.

Unit V. Code optimization:
Consideration for Optimization, Scope of Optimization, local optimization, loop optimization, frequency reduction, folding, DAG representation.
Data flow analysis:
Flow graph, data flow equation, global optimization, redundant sub expression elimination, Induction variable elements, Live variable analysis, Copy propagation.
Object code generation:
Object code forms, machine dependent code optimization, register allocation and assignment generic code generation algorithms, DAG for register allocation.

Text Book

1. Compilers- Principles, Techniques and Tools, By A.V. Aho, M.S. Lam, R Sethi and J.D.Ullman, Pearson Education.

Reference Book

1. lex &yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Engineering a Compiler, by Cooper & Linda, Elsevier.
3. Compiler Construction, K.C. Louden, Thomson Brooks/Cole.

MCA-ECA306 Parallel Computing Cr 4

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

MCA-ECA306.CO1.	Understand the basic concepts, terminology, and models of parallel computing, including PRAM model, data parallelism, control parallelism, and scalability.
MCA-ECA306.CO2.	Explain the difference between various parallel computing approaches such as pipelining, data parallelism, and control parallelism, and their relevance to modern computational demands.
MCA-ECA306.CO3.	Apply parallel computing algorithms (e.g., prefix sums, list ranking, matrix multiplication) to solve computational problems effectively using parallel processing techniques.
MCA-ECA306.CO4.	Analyze the efficiency and scalability of parallel algorithms, comparing sequential versus parallel solutions in terms of time complexity and resource utilization.
MCA-ECA306.CO5.	Evaluate and select appropriate parallel algorithms and computational models (e.g., MIMD, PRAM) for solving problems in linear systems, matrix multiplication, and other computational tasks.

Unit I. Introduction

Computational Demand of Modern Science, Advent of Practical Processing, Parallel Processing Terminology- Contrasting Pipelining and Data Parallelism, Control Parallelism, Scalability, Control- Parallel Approach, Data-Parallel Approach with I/O.

Unit II. PRAM Algorithm

A Model of Serial Computation, The PARAM Model of Parallel Computation, PARAM Algorithm- Parallel Reduction, Prefix Sums, List Ranking, Preorder Tree Traversal, Merging Two Sorted Lists, Graph Colouring, Problem defining Fast Solutions on PRAMS.

Unit III. Elementary Parallel Algorithm

Classifying MIMD Algorithm, Reduction.

Unit IV. Matrix Multiplication

Sequential Matrix Multiplication, Algorithms for Processor Array, Algorithms for Multiprocessors.

Unit V. Solving Linear Systems

Terminology, Back Substitution, ODD-EVEN Reduction, Gaussian Elimination, The JACOBI Algorithm, The Gauss-Seidel Algorithm, Jacobi Overrelaxation and Successive Overrelaxation, Mulyigrid Methods, Conjugate Gradient.

Text Book and References:

1. H.Attiya & J. Welch- Distributed Computing- Fundamentals, Simulations and Advanced Topics, 2ndEdn., Wiley India Publication, New Delhi, 2006.

2. M.J. Quinn-Parallel Computing-Theory and Practice, 2nd Edn. McGraw Hill Inc., New York.

MCA-ECA307 Cloud Computing Cr 4

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

MCA-ECA307.CO1	Understand the fundamental concepts, characteristics, and models of cloud computing, including various deployment and service models such as IaaS, PaaS, and SaaS.
MCA-ECA307.CO2	Explain the underlying principles of cloud computing, including distributed systems, grid computing, and virtualization, and how they contribute to the cloud environment.
MCA-ECA307.CO3	Apply cloud computing services such as resource provisioning, virtualization, and scheduling techniques for managing cloud infrastructure and applications.
MCA-ECA307.CO4	Analyze cloud-based applications and the challenges associated with cloud computing, including security, privacy, and integration with other technologies like wireless sensor networks.
MCA-ECA307.CO5	Evaluate different cloud computing platforms and services (e.g., Google App Engine, Microsoft Azure, Salesforce) for suitability in various business applications and solve real-world cloud computing problems.

Unit I. Introduction

Introduction to Cloud Computing, Roots of Cloud Computing: Fundamental concepts of DistributedSystems, Cluster Computing, Grid Computing, and Mobile Computing.

Unit II. Cloud Models

Basics of Cloud Computing Concepts, Characteristics of Cloud Computing, Need for Cloud, Cloud Deployment models: private, public, hybrid and community cloud, Cloud Services: Resource-as-a- Service (RaaS), Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a- Service (PaaS), Examples of each services.

Unit III. Cloud Services

RaaS: Usage of Physical resources like servers, networks, data centered, IaaS: Virtualization, Virtual Machine provisioning and Migration Services, Scheduling techniques of Virtual machines for resource reservation. PaaS: Integrated lifecycle platform: Google App Engine, Microsoft Azure, Anchored life cycle platform: Salesforce platform, SaaS: Characterizing SaaS, Salesforce’s software environment.

Unit IV. Cloud Application

Cloud Application, Cloud challenges, Cloud Security and privacy issues, Mobile Cloud, Integration of Cloud with Wireless Sensor Network and its application.

Unit V. Cloud Security and Management

Security challenges in cloud computing, Security concerns in cloud environments: Data breaches, loss of data control, and data availability, Privacy issues and legal aspects of cloud computing, Protecting sensitive data and compliance regulations (e.g., GDPR, HIPAA), Resource management in cloud environments, Virtual resource management, Cloud load balancing techniques.

Text Book and References:

1. “Cloud Computing Principles and Paradigms”, edited by Rajkumar Buyya, James Broberg andAndrzej Goscinski, Wiley Publication.
2. “Fundamentals of Cloud Computing”, P.K. Pattnaik, M.R. Kabat, S. Pal, Vikas Publishing, 2014.
3. “Cloud Computing for Dummies”, Judith Hurwitz, Robin Bloor, Marcia Kaufman and FernHalper, Wiley Publication.
4. “New frontiers in information and software as a service”, Divyakant Agrawal, K. SelcukCandan,WenSyan Li (Eds.), Springer Proceedings.

MCA-ECA308 Bio-informatics Cr 4

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

MCA-ECA308.CO1.	Describe key concepts in bioinformatics, including biomedical data types, clinical databases, patient monitoring systems, and electronic health records (EHR).
MCA-ECA308.CO2.	Understand the principles of bioinformatics and their applications in clinical decision-making, evidence evaluation, and the integration of health information systems.
MCA-ECA308.CO3.	Apply bioinformatics methods such as Bayesian networks and machine learning techniques for medical data analysis and decision-making in healthcare environments.
MCA-ECA308.CO4.	Analyze clinical databases, healthcare informatics systems, and evaluate the effectiveness of various decision support systems in healthcare delivery.
MCA-ECA308. CO5.	Design and implement bioinformatics algorithms, decision-making systems, and clinical databases that address healthcare issues and improve medical practices and patient care.

Unit I. Introduction

Biomedical data,-Clinical and life sciences -standards and databases. Principles and its uses Electronic health records (EMR) and health Information exchanges—including information retrieval, medical decision making, evaluation and evidence. Patient monitoring systems-ethics in informatics - bayesian networks-learning and decision-data structure in algorithm design and analysis.

Unit II. Networking

TCP/IP Sockets and DNS clinical database concepts-design of the clinical information systems/Clinical Decision support systems--Synchornization, concurrency, deadlock, full-text databases, distributed database services and architexture on one of the database.any clinical database structure as one example.

Unit III. Methods and Evaluation

Sampling, appropriate use of controls, data collection including human-testing of statiscal significance, sensitivity and specificity.ROC plots. Methods and issues specific to healthcare.

Unit IV. Healthcare informatics

Understanding and interaction Health organization especially academic health centers, understanding the health care environment, understanding the organization informatics-Interaction between these three units-machine learning approaches to make decision making and discovery. Human factors in clinical systems –use of machine learning to make modeling, datamining, policy design and law. Translation research and its uses and implications Evidence based medicines.

Unit V. Advanced Bioinformatics Techniques and Applications

Introduction to genomic and proteomic data, tools for sequence alignment, gene expression analysis, and functional genomics.
Bioinformatics Algorithms: Advanced algorithms for sequence comparison, motif discovery, and secondary structure prediction.
Big Data in Bioinformatics: Challenges and strategies for handling large-scale biological datasets, cloud computing in bioinformatics.

Text Book and References:

1. Biomedical Informatics: First edition, - By Jules J. Berman. Jones & Bartlett, 2010
2. Biomedical Informatics: computer applications in Health care and Biomedicine (3rd ed), by Shortliffe EH, Ciminio JJ., New York Springer-Verlag. 2000
3. Evaluation methods in medical Informatics by Friedman CP. Wyatt JC, New York Springer-Verlag-1996.

MCA-ECA309 Internet of Things Cr 4

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

MCA-ECA309.CO1	Explain the basic concepts of IoT, including its definition, characteristics, functional blocks, and communication protocols.
MCA-ECA309.CO2	Explain the differences between M2M (Machine-to-Machine) communications and IoT, and describe the evolving industrial structure for IoT integration.
MCA-ECA309.CO3	Apply IoT concepts by designing and implementing IoT solutions using platforms like Arduino, Raspberry Pi, and cloud computing technologies.
MCA-ECA309.CO4	Analyze the various architectural models of IoT, including functional, information, operational, and deployment views, and evaluate the design constraints affecting IoT solutions.
MCA-ECA309.CO5	Design and develop domain-specific IoT applications (e.g., home automation, industry, surveillance) by utilizing IoT tools and platforms, addressing privacy and security concerns in the process.

Unit I. Introduction to IoT:
Sensing, Actuation, Networking basics, Communication Protocols, Sensor Networks, Machine-to-Machine Communications, IoT Definition, Characteristics. IoT Functional Blocks, Physical design of IoT, Logical design of IoT, Communication models & APIs.

Unit II. M2M to IoT - The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics. Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT.

M2M vs IoT an Architectural Overview – Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. Reference Architecture and Reference Model of IoT.

Unit III. IoT Reference Architecture - Getting Familiar with IoT Architecture, Various architectural views of IoT such as Functional, Information, Operational and Deployment. Constraints affecting design in IoT world- Introduction, Technical design Constraints.

Unit IV. Domain specific applications of IoT: Home automation, Industry applications, Surveillance applications, Other IoT application.

Unit V. Developing IoT solutions: Introduction to Python, Introduction to different IoT tools, Introduction to Arduino and Raspberry Pi Implementation of IoT with Arduino and Raspberry, Cloud Computing, Fog Computing, Connected Vehicles, Data Aggregation for the IoT in Smart Cities, Privacy and Security Issues in IoT.

Text Book:

1) Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-onApproach)”, 1st Edition, VPT, 2014.

References Books:

1) SudipMisra, Anandarup Mukherjee,Introduction to IoT,Cambridge University Press, 2022
2) Rashmi Nanda, IoT and Smart Cities: Your smart city planning guide, BPB Publications, 2022
3) Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013.
4) Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014.
5) CunoPfister, Getting Started with the Internet of Things, O‟Reilly Media, 2021, ISBN: 978-1-4493- 9357-1.

MCA-ECA310 Theory of Computation Cr 4

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

MCA-ECA310.CO1.	Understand the fundamental mathematical concepts related to sets, functions, relations, alphabets, and strings, and their application to formal languages and automata theory.
MCA-ECA310.CO2.	Explain the core concepts of finite automata (DFA, NFA), regular languages, regular expressions, and the capabilities and limitations of these automata models.
MCA-ECA310.CO3.	Apply mathematical induction, pumping lemmas, and algorithms to solve problems involving regular and context-free languages and automata (e.g., DFA, NFA, PDA).
MCA-ECA310.CO4.	Analyze the properties of context-free grammars, pushdown automata, and Turing machines, and evaluate their equivalence in terms of language recognition and computation.
MCA-ECA310.CO5.	Evaluate the computational power of Turing machines and recursive enumerable languages, applying decision algorithms, and understanding the limits of computation, such as the Church-Turing thesis.

Unit I. Basic Mathematical Objects and Mathematical Induction
Sets, logic, Functions, Relations, Alphabets, Strings, Languages, Principle of mathematical induction,Recursive definition.

Regular Expressions and Finite Automata
Regular languages and Regular Expressions, Memory required to recognize a language, Finite Automata, capability & limitations of FSM, Deterministic Finite Automata , Non-Deterministic Finite Automata, NFA with e-moves, regular sets & regular expressions, Equivalence of DFA and NDFA, NFA from regular expressions, regular expressions from DFA, Moore versus Mealy m/c, two way finite automata equivalence with one way , Kleen’s Theorem, applications of finite automata.

Unit II. Regular and Non-regular languages
Criterion for Regularity, Minimal Finite Automata, Pumping Lemma for Regular Languages, Decisionproblems, Regular Languages and Computers.

Unit III. Context Free Grammars
Introduction, definition, Regular Grammar, Derivation trees, Ambiguity, Simplified forms and NormalForms, Applications.

Unit IV. Pushdown Automata
Definition, Moves, Instantaneous Descriptions, Language recognised by PDA, Deterministic PDA, Acceptance by final state & empty stack, Equivalence of PDA , Pumping lemma for CFL, Interaction andComplements of CFL,Decision algorithms.

Unit V. Turing Machines
Definition and examples, Computing Partial Functions with Turing Machine(TM), Combining TMs,

Variations of TMs, Multi-tape TMs, Non-deterministic TM, Universal TM, Church Thesis.

Recursively Enumerable Languages

Recursively Enumerable and Recursive, Enumerating Language, Context Sensitive and Chomosky Hierarchy

Text Book and References:

1. John Martin -"Introduction to Languages and the Theory of Computation", 3rd edition, TMH.
2. K.L.P Mishra & N. Chandrasekharan -"Theory of Computer Science", PHI
3. Hopcroft JE. And Ullman JD -"Introduction to Automata Theory, Languages & Computation",Narosa.
4. Lewis H. R. and Papadimitrou C. H -"Elements of the theory of Computation", PHI.

MCA-ECA311 Data Mining and Data Warehousing Cr 4

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

MCA-ECA311.CO1.	Understand the functionality of the various data mining and data warehousing component.
MCA-ECA311.CO2.	Appreciate the strengths and limitations of various data mining and data warehousing models.
MCA-ECA311.CO3.	Explain the analyzing techniques of various data
MCA-ECA311.CO4.	Describe different methodologies used in data mining and data ware housing.
MCA-ECA311.CO5.	Compare different approaches of data ware housing and data mining with various technologies.

Unit I. Introduction

Introduction, Data Mining as subject, What motivated Data mining, Why is it important.

Data warehousing

Introduction, What is a Data Warehousing Definition, Multidimensional Data Model, OLAP Operation, warehouse Scheme, Data Warehousing, Architecture, Metadata, OLAP ENGINE, Data warehouse Backend Process.

Data Mining

Introduction, What is Data Mining, Data Mining Definition, KDD Vs, Data Mining, DBMS Vs. DM, Other related area, DM Technique, Other Mining Problem, Issue and challenge is in DM, DM Application area, DM Application, Case Sturdy.

Unit II. Mining Association Rule in Large Database

Introduction, what is an Association Rule, Method to discover association Rule, A Priori Algorithm, Partition Algorithm, Pinear- Search algorithm, Dynamic item set Counting Algorithm, FP – Tree Growth Algorithm, Discussion and Different Algorithm, Generalized, Association Rule, Association Rules with Item Constraints.

Unit III. Clustering Techniques

Introduction , Clustering Paradigram, Partition Algorithm, K- Medoid Algorithm, CLARA, CLARANS, Heirer Chial Clustering, DBSCAN, BIRCH, CURE, Categorical Chastereing Algorithms, STIRR, ROCK, CACTUS.

Unit IV. Data Mining Primitives, Language and System Architecture

Data Mining Primitives, what defines a Data Mining task, Task relevant Data, The Kind of Knowledge to Mined, Concept Hicrechings, Interestingness Measwce, presenation and visualization of DiscoveredPatterns, Data Mining Query Language.

Unit V. Decision Trees

Introductions, What is decision Tree, Tree Construction Principle, Best split splitting Indices, Splitting criteria, Decision Tree Construction with Presenting, Prunesing Technique, Integration of Pruning Technique and Construction.

Temporal and Spatial Data Mining

Introduction, What is Temporal Data Mining emporal Association Rules, Sequence Mining, The

GSPAlgorithm, SPIRIT, Spafial Mining, Spatial Clustering, Spatial Trends.

Text Book and References:

1. A.K. Pujari, “A Data Mining Technique”, University press (India) Limited, 2001
2. A Hand and M. Kamber, “Data Mining Concept and Technique”, Morgan Kauffmann Publishers, Else river India, New Delhi, 2003.
3. Recherd J, Roiger and Michance W. Creatz, Data Mining: a tutorial Based Primer, Addision Wesley,2003.
4. M.H. Dienham, Data Mining : Introductory and Advanced Topics, Pentice Hall 2003.

MCA-ECA312 Deep Learning Cr 4

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

MCA-ECA312.CO1	Describe the key concepts and techniques in deep learning, including paradigms, frameworks, neural networks, activation functions, and loss functions.
MCA-ECA312.CO2	Explain the workings of feed-forward neural networks, backpropagation, regularization methods, and optimization techniques used in training deep learning models.
MCA-ECA312.CO3	Apply deep learning techniques to train and optimize various models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for real-world applications.
MCA-ECA312.CO4	Analyze and compare different deep learning models, including Conditional Random Fields (CRFs), Deep Belief Networks, and Probabilistic Neural Networks (PNNs), and evaluate their suitability for various tasks.
MCA-ECA312.CO5	Design and implement deep learning solutions using advanced techniques like autoencoders, sparse coding, and object recognition, and use deep learning tools such as Caffe, Theano, and Torch for model development and research.

Unit I. Introduction

Various paradigms of earning problems, Perspectives and Issues in deep learning framework, review offundamental learning techniques.

Unit II. Feed forward neural network

Artificial Neural Network, activation function, multi-layer neural network

Training Neural Network

Risk minimization, loss function, backpropagation, regularization, model selection, and optimization

Unit III. Conditional Random Fields

Linear chain, partition function, Markov network, Belief propagation, Training CRFs, Hidden MarkovModel, Entropy

Unit IV. Deep Learning

Deep Feed Forward network, regularizations, training deep models, dropouts, Convolutional NeuralNetwork, Recurrent Neural Network, Deep Belief Network

Unit V. Probabilistic Neural Network

Hopfield Net, Boltzman machine, RBMs, Sigmoid net, Autoencoders

Deep Learning research

Object recognition, sparse coding, computer vision, natural language processing

Deep Learning Tools

Caffe, Theano, Torch

Text Book and References:

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016
2. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006
3. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009
4. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013

ELECTIVE COURSES (SET B)

MCA-ECB301 Mobile Computing Cr 3

Course Outcome: At the end of the course, the student should be able to:

MCA-ECB301.CO1	Understanding the basics of mobile telecommunication system
MCA-ECB301.CO2	Apply the required functionality at each layer for given application
MCA-ECB301.CO3	Analyze solution for each functionality at each layer
MCA-ECB301.CO4	Identify simulator tools and design Ad hoc networks
MCA-ECB301.CO5	Develop a mobile application.

Unit I. Introduction

Mobile Computing – Mobile Computing vs wireless Networking – Mobile Computing Applications – Characteristics of Mobile computing – Structure of Mobile Computing Application. MAC Protocols – Wireless MAC Issues – Fixed Assignment Schemes – Random Assignment Schemes – Reservation Based Schemes.

Unit II. Mobile internet protocol and transport layer

Overview of Mobile IP – Features of Mobile IP – Key Mechanism in Mobile IP – route Optimization. Overview of TCP/IP – Architecture of TCP/IP- Adaptation of TCP Window – Improvement in TCP Performance.

Unit III. Mobile telecommunication system

Global System for Mobile Communication (GSM) – General Packet Radio Service (GPRS) – Universal Mobile Telecommunication System (UMTS).

Unit IV. Mobile ad-hoc networks

Ad-Hoc Basic Concepts – Characteristics – Applications – Design Issues – Routing – Essential of Traditional Routing Protocols –Popular Routing Protocols – Vehicular Ad Hoc networks (VANET) – MANET Vs VANET – Security.

Unit V. Mobile platforms and applications

Mobile Device Operating Systems – Special Constrains & Requirements – Commercial Mobile Operating Systems
– Software Development Kit: iOS, Android, BlackBerry, Windows Phone – M Commerce – Structure – Pros &Cons – Mobile Payment System – Security Issues.

Text Book and References:

1. Rajib Mall, “Fundamentals of Mobile Computing”, PHI Learning Pvt. Ltd, New Delhi – 2016, second edition.
2. C.K.Toh, “AdHoc Mobile Wireless Networks”, First Edition, Pearson Education, 2002.
3. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober, “Principles of Mobile Computing”,Springer, 2003.

MCA-ECB302 Windows Programming with Visual Basic.Net Cr 3

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

MCA-ECB302.CO1	Explain fundamental concepts of Visual Basic .NET, including the Integrated Development Environment (IDE), control classes, data types, variables, and decision-making constructs.
MCA-ECB302.CO2	Explain the process of creating Windows forms and applications in VB.NET, including the usage of object-oriented programming principles such as procedures, classes, and inheritance.
MCA-ECB302.CO3	Apply VB.NET programming techniques to develop Windows-based applications, including MDI (Multiple Document Interface) applications, file input/output, and multi-threaded applications.
MCA-ECB302.CO4	Analyze and troubleshoot common errors in VB.NET applications through exception handling, debugging tools, and creating custom error reports.
MCA-ECB302.CO5	Design, develop, and deploy VB.NET applications with features such as custom user controls, database interaction using ADO.NET, and integration with web services and Crystal Reports.

Unit I. Introduction to Visual Studio. NET, IDE to develop VB. NET ,Control Classes, Different type of applications., creating Windows form, Data types, Variables, Literals, decision making, Looping constructs, Object- oriented Programming through VB. NET, Creating procedures, Using Commondialogue classes, Retrieving and manipulating data to store in data base by ADO. NET..

Unit II. Creating MDI applications, Creating Menus, Performing File Input/ Output. Creating multi- threadedapplications.

Unit III. Handling exception, debugging application, creating application assistant from help System, creating and using components, creating user control.

Unit IV. Generating a report using crystal report.

Unit V. Creating and using web-service, deploying an application.

Text Book and References:

1. Visual Basic.Net Programming Black Book, Steven Holzner, Dreamtech Press.
2. Professional Visual Basic.Net 2003, Bill Evjen, Willey Dreamtech Press.

MCA-ECB304 System Administration and Linux Cr 3

MCA-ECB304.CO1	Understand the role and responsibilities of a Linux system administrator.
MCA-ECB304.CO2	Install and configure the Linux operating system.
MCA-ECB304.CO3	Manage the resources and security of a computer running Linux at a basic level.
MCA-ECB304.CO4	Make effective use of Linux utilities, and scripting languages.
MCA-ECB304.CO5	Configure and mange simple TCP/IP network services on a Linux system.

Unit I: Introduction to Linux Operating System
Overview of Linux: History, evolution, and different distributions (Ubuntu, CentOS, Red Hat, etc.).
Linux Architecture: Kernel, Shell, File system hierarchy, and processes.
Linux Installation: Installation of Linux (dual boot, virtual machine setup), Partitioning, file systems (ext4, xfs).
Command Line Basics: Introduction to terminal, file management commands (ls, cp, mv, rm, etc.), text manipulation tools (cat, grep, wc, etc.).
Shell Scripting: Basics of shell programming, writing simple shell scripts, variables, loops, conditions.

Unit II: Linux File System and Disk Management
Linux File System: File system structure, inodes, mounting and unmounting filesystems.
Disk Partitioning: Tools like fdisk, parted, creating and managing partitions.
File System Management: File system types, formatting file systems, checking file system integrity (fsck).
Backup and Restoration: Backup tools (tar, rsync), compression techniques, backup strategies, restoring data.
Disk Quotas: Managing disk space, setting up user quotas.

Unit III: User Management and Permissions
User Management: Creating and managing users and groups, setting passwords, user home directories.
File Permissions: Understanding read, write, and execute permissions for users, groups, and others, chmod, chown, and chgrp commands.
Access Control: Using ACLs (Access Control Lists), setting file permissions, understanding special permissions (setuid, setgid, sticky bit).
Sudo and Root Privileges: Configuring sudo, managing root privileges, and understanding security implications of sudo.
Security Best Practices: Securing user accounts, disabling unused accounts, managing password policies.

Unit IV: Networking and Services Configuration
Networking Basics: TCP/IP fundamentals, configuring IP addresses, netmask, routing.
Network Configuration: Using ifconfig, ip command, configuring network interfaces, static and dynamic IP addressing.
Network Services: Setting up essential services like SSH, FTP, DNS, and DHCP.
Firewall and Security: Introduction to firewall concepts, configuring iptables/firewalld, securing a Linux system with firewall rules.

System Logging: Managing system logs using syslog and journald, configuring log rotation, analyzing log files.

Unit V: Process Management, System Monitoring, and Troubleshooting
Process Management: Viewing and managing processes using ps, top, kill, and nice commands, process scheduling.
System Monitoring: Tools for monitoring system performance (htop, iostat, vmstat), analyzing CPU, memory, and disk usage.
Package Management: Installing, updating, and removing software packages using package managers like apt, yum, dnf, and zypper.
System Troubleshooting: Identifying and resolving system issues, understanding system logs, network troubleshooting, diagnosing hardware problems.
System Maintenance: System updates, kernel upgrades, and basic system health checks.

Text Book and References:

1. Linux Administration: A Beginner’s Guide, Shah, TMH.
2. LINUX: The Complete Reference, Petersen, TMH.
3. Guide to LINUX installations & administration, Wealls, VIKAS.
4. Red Hat LINUX- Administrator’s Guide, Cox, PHI.
5. LINUX Network Administrator’s Guide, Kirch, SPD/O’REILLY.

MCA-ECB305 Image Processing Cr 3

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

MCA-ECB305.CO1	Analyze different image processing technique to retrieve image information.
MCA-ECB305.CO2	Differentiate between different image transformation techniques.
MCA-ECB305.CO3	Analyze different image enhancement techniques.
MCA-ECB305.CO4	Analyze the concept of color image processing and image restoration.
MCA-ECB305.CO5	Differentiate between different image compression and segmentation techniques.

Unit I. Introduction: Digital Image representation; Fundamental steps in Image processing, Elements of digital Image processing systems.
Digital Image Fundamentals: Sampling and quantization, Imaging geometry.

Unit II. Image Transforms: Fourier, Walsh, Hademord, discrete cosine and Hotelling transforms and their properties.

Unit III. Image Enhancement: Enhancement by point processing, spatial filtering, Frequency domain enhancement, Color image processing.
Image Restoration: Unconstrained and constraint restoring, inverse filtering, Wiener Filter, Geometric transforms.

Unit IV. Image Compression: Image Compression models, Error-free compression, Lossy compression, Image compression standards.

Unit V. Image Segmentation: Detection of discontinuities, edge linking, Thresholding.
Representations and Descriptions: Chain codes, shape numbers, moments and Fourier and other descriptors. **Recognition & Interpretations;**

Text Book and References:

1. R.C. Gonzalez & R./E. Woods, Digital Image Processing : Addison - Wesley Pub. Comp
2. Ralph Gonzalez, Richard Woods, Steven Eddins, Digital Image Processing Using MATLAB, McGraw Hill Education (India) Private Limited; 2 edition (8 June 2010)
3. Digital image Processing &Analysis, Chanda&Magumde, PHI.
4. Fundamentals of Digital Image Processing, Jain, PHI.
5. Image Processing, Analysis & Machine Vision, Sonka, VIKAS.

MCA-ECB306 Advanced Python Programming Cr 3

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

MCA-ECB306.CO1	Understand the advanced Python programming concepts including file handling, exception handling, and advanced data structures such as dictionaries, sets, and lists.
MCA-ECB306.CO2	Explain the key features of object-oriented programming in Python, such as inheritance, polymorphism, encapsulation, and abstract classes, and their applications in real-world scenarios.
MCA-ECB306.CO3	Apply Python libraries such as NumPy, Pandas, and Matplotlib to solve complex computational problems, manipulate data, and visualize results.
MCA-ECB306.CO4	Analyze and optimize Python code for performance using advanced features such as decorators, generators, and multithreading, and debug code using effective error-handling techniques.
MCA-ECB306.CO5	Design and implement Python-based applications by integrating advanced features like web scraping, API interaction, and machine learning algorithms, showcasing problem-solving and application development skills.

Unit I. Introduction to Python

Introduction to Python programming, use IDE to develop programs, Basic coding skills, working with data types and variables, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure, working with lists, work with a list of lists, work with tuples, work with dates and times, get started with dictionaries.

Unit II. Classes in Python

OOPS Concepts, Classes and objects , Classes in Python, Constructors, Data hiding, Creating Classes, Instance Methods, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes, Iterators, generators and decorators.

Unit III. I/O and Error Handling In Python

Introduction, Data Streams, Creating Your Own Data Streams, Access Modes, Writing Data to a File, Reading Data From a File, Additional File Methods, Handling IO Exceptions, Errors, Run Time Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions, Working withDirectories.

Unit IV. An Introduction to relational databases

SQL statements for data manipulation, Using SQLite Manager to work with a database, Using Python to work with a database, Creating a GUI that handles an event, working with components.

Unit V. Implement Machine Learning algorthims:

Usage of Numpy for numerical Data, Usage of Pandas for Data Analysis, Matplotlib for Python plotting, Seaborn for Statistical plots, interactive Dynamic visualizations, SciKit for Machine learning.

TEXT BOOKS

1. Michael Urban and Joel Murach, Python Programming, Shroff/Murach,2016
3. Mark Lutz, Programming Python, O`Reilly, 4th Edition,2010.

MCA-EPB301 Mobile Computing Lab Cr 2

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

MCA-EPB301.CO1	Understand the fundamental concepts in mobile computing, including mobile networks, protocols, and mobile operating systems.
MCA-EPB301.CO2	Describe the principles of mobile communication technologies, such as GSM, GPRS, CDMA, and LTE, and their role in mobile applications.
MCA-EPB301.CO3	Implement basic mobile network communication protocols and demonstrate their practical use through hands-on projects such as mobile app development and device communication.
MCA-EPB301.CO4	Analyze and debug mobile computing applications, identify performance bottlenecks, and optimize solutions for mobile environments.
MCA-EPB301.CO5	Design and develop mobile applications using development environments such as Android Studio, Xcode, and other mobile SDKs, incorporating essential features such as UI/UX design and network communication.

List of Experiments

Experiment 1: Install and configure Android Studio and create a simple "Hello World" mobile application.
Experiment 2: Implement a basic mobile application that uses user input through buttons, text fields, and basic event handling.
Experiment 3: Develop a mobile app to retrieve and display data from a web API (JSON or XML).
Experiment 4: Create a mobile app that communicates over Bluetooth with another device, sending and receiving text messages.
Experiment 5: Design and develop a mobile application that uses GPS to display the user’s current location on a map.
Experiment 6: Implement GPRS/3G-based communication in a mobile app, such as fetching data from a server over HTTP.
Experiment 7: Develop a simple chat application using Socket programming for mobile communication.
Experiment 8: Create a mobile app that uses the SQLite database to store and retrieve data locally on the mobile device.
Experiment 9: Implement push notifications in a mobile app using Firebase or any other notification service.
Experiment 10: Design and develop an application that interacts with a remote server using RESTful web services.

MCA-EPB302 Windows Programming with Visual Basic.Net Cr 2

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

MCA-EPB302.CO1	Describe the basic concepts of Windows programming using Visual Basic.Net, including the .NET framework, VB.NET syntax, and control structures.
MCA-EPB302.CO2	Understand the core concepts of object-oriented programming (OOP) in VB.NET, such as classes, objects, inheritance, and polymorphism, and their applications in Windows applications.
MCA-EPB302.CO3	Apply Visual Basic.NET programming skills to develop interactive Windows applications with graphical user interfaces (GUIs) using various controls and components.
MCA-EPB302.CO4	Analyze and debug VB.NET programs to identify and fix issues related to user input validation, data manipulation, and error handling in Windows applications.
MCA-EPB302.CO5	Design and implement complex Windows-based applications using VB.NET, incorporating advanced features like multi-threading, database integration, and reporting.

List of Experiments

1. Introduction to Visual Basic .NET.
2. User Interface Design.
3. Variables, Constants and Calculations
4. Decisions making, Looping.
5. Arrays
6. Forms and Controls, MDI Forms, Common Dialog Boxes.
7. Sub Procedures, Functions.
8. Parameter passing by value and by reference.
9. Thread/Time slicing related programs.
10. Class /Interface based programming (Object Oriented Programs).
11. Inheritance / Polymorphism.
12. Database connectivity.
13. Saving Data and Objects in Files.
14. Web Forms.

MCA-EPB304 System Administration and Linux Lab Cr 2

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

MCA-EPB304.CO1	Understand the basic concepts of Linux operating system, file systems, and essential commands used in system administration.
MCA-EPB304.CO2	Understand the Linux system architecture, user management, file permissions, and system processes and their role in system administration.
MCA-EPB304.CO3	Apply Linux system administration skills to configure user accounts, manage files and directories, and perform basic networking tasks using command-line tools.
MCA-EPB304.CO4	Analyze system logs, monitor system performance, and troubleshoot common issues such as system crashes, service failures, and network problems.
MCA-EPB304.CO5	Design and implement a basic system administration workflow in Linux, including configuring servers, managing security, and automating tasks using shell scripts.

List of Experiments

Experiment 1: Install a Linux operating system (e.g., Ubuntu or CentOS) and perform initial configuration tasks like setting the hostname, network configuration, and creating user accounts.

Experiment 2: Create and manage user accounts and groups in Linux. Use commands like `useradd`, `groupadd`, `passwd`, and `usermod` to perform administrative tasks.

Experiment 3: Configure file permissions, ownership, and file attributes in Linux using commands such as `chmod`, `chown`, `chgrp`, and `umask`.

Experiment 4: Demonstrate process management in Linux by using commands such as `ps`, `top`, `kill`, and `nice` to monitor and manage system processes.

Experiment 5: Perform disk partitioning, mounting, and managing file systems using tools like `fdisk`, `mkfs`, and `mount`.

Experiment 6: Configure basic network settings in Linux, including IP configuration, testing network connectivity using `ifconfig`, `ping`, `netstat`, and setting up network services (e.g., DNS, DHCP).

Experiment 7: Set up and configure a basic Apache HTTP server or Nginx web server, ensuring it is running and serving web pages.

Experiment 8: Implement system backups using `tar` and `rsync`, and automate backup processes using cron jobs.

Experiment 9: Monitor system performance using tools such as `vmstat`, `iostat`, and `netstat`, and interpret logs from `/var/log` to troubleshoot system performance.

Experiment 10: Write and execute simple shell scripts to automate administrative tasks like log rotation, user management, or disk usage monitoring.

MCA-EPB305 Image Processing Lab Cr 2

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

MCA-EPB305.CO1	Describe the fundamental concepts and techniques of image processing, including image representation, transformations, and enhancement methods.
MCA-EPB305.CO2	Understand the basic image processing operations such as image filtering, enhancement, and edge detection, and their application to real-world problems.
MCA-EPB305.CO3	Apply various image processing techniques, including noise reduction, contrast enhancement, and morphological operations using appropriate software tools and libraries.
MCA-EPB305.CO4	Analyze image data, identify problems such as noise, blurring, and distortion, and apply appropriate techniques to restore or enhance the image quality.
MCA-EPB305.CO5	Evaluate the performance of image processing algorithms based on metrics such as quality improvement, processing time, and computational efficiency, and compare different methods to select the most effective one for a given task.

(Simulation tool: Matlab/ Python)

1. Simulation and Display of an Image, Negative of an Image(Binary & Gray Scale)
2. Implementation of Relationships between Pixels
3. Implementation of Transformations of an Image
4. Contrast stretching of a low contrast image, Histogram, and Histogram Equalization
5. Display of bit planes of an Image
6. Display of FFT(1-D & 2-D) of an image
7. Computation of Mean, Standard Deviation, Correlation coefficient of the given Image
8. Implementation of Image Smoothening Filters(Mean and Median filtering of an Image)
9. Implementation of image sharpening filters and Edge Detection using Gradient Filters
10. Image Compression by DCT,DPCM, HUFFMAN coding
11. Implementation of image restoring techniques
12. Implementation of Image Intensity slicing technique for image enhancement
13. Canny edge detection Algorithm

MCA-EPB306 Advanced Python Programming Lab Cr 2

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

MCA-EPB306.CO1	Write advanced Python programming concepts such as object-oriented programming (OOP), decorators, generators, and working with libraries for web scraping, data manipulation, and visualization.
MCA-EPB306.CO2	Understand how to implement advanced Python techniques such as multithreading, multiprocessing, and asynchronous programming to optimize performance and handle concurrent tasks.
MCA-EPB306.CO3	Apply Python libraries like NumPy, pandas, and matplotlib to process and analyze large datasets, visualize data, and build data-driven applications.
MCA-EPB306.CO4	Analyze complex problems and design Python-based solutions using appropriate algorithms, data structures, and Python modules for real-world applications in areas like machine learning, web development, or automation.
MCA-EPB306.CO5	Create Python-based applications by integrating various advanced concepts such as web development (using Flask/Django), database interactions, and implementing RESTful APIs for seamless user interaction and backend processing.

LIST OF PROGRAMS:

1) **Exception Handling and User defined exception(s)**

a) Write a python program to catch following exception

i) Value Error

ii) Index Error

iii) Name Error

iv) Type Error

v) DivideZero Error

b) Write a python program to create user defined exceptions.

c) Write a python program to understand the use of else and finally block with try block.

d) Write a python program that uses raise and exception class to throw an exception.

2) **Modules and Packages**

a) Write a python program to create a module and import the module in another python program.

b) Write a python program to import all objects from a module, specific objects from module and provide custom import name to the imported object from the module.

c) Create a python package having at least two modules in it.

d) Create a python package having at least one subpackage in it.

3) **Numpy Library**

a) Create a numpy array from list, tuple with float type

b) Python program to demonstrate slicing, integer and boolean array indexing

c) Write a python program to find min, max, sum, cumulative sum of array.

d) Write a python program to demonstrate use of ndim, shape, size, dtype.

4) **Numpy Library: Linear Algebra**

a) Write a python program to find rank, determinant, and trace of an array.

b) Write a python program to find eigenvalues of matrices

c) Write a python program to find matrix and vector products (dot, inner, outer, product), matrix exponentiation.

d) Write a python program to solve a linear matrix equation, or system of linear scalar equations.

5) **Numpy Advanced**

a) Create a white image using NumPy in Python and

b) Convert a NumPy array to an image and Convert images to NumPy array?

c) Perform Sorting, Searching and Counting using Numpy methods.

d) Write a program to demonstrate the use of the reshape() method.

6) **Pandas Library**

a) Write a python program to implement Pandas Series with labels.

b) Create a Pandas Series from a dictionary.

c) Creating a Pandas DataFrame.

d) Write a program which make use of following Pandas methods

i) describe()

ii) head()

iii) tail()

7) **Pandas Library: Selection**

a) Write a program that converts Pandas DataFrame and Series into numpy.array.

b) Write a program that demonstrates the column selection, column addition, and column deletion.

c) Write a program that demonstrates the row selection, row addition, and row deletion.

d) Get n-largest and n-smallest values from a particular column in Pandas dataFrame

8) **Pandas Library: Visualization**

a) Write a program which use pandas inbuilt visualization to plot following graphs:

i. Bar plots

ii. Histograms

iii. Line plots

iv. Scatter plots

b) Write a program to demonstrate use of groupby() method.

c) Write a program to demonstrate pandas Merging, Joining and Concatenating

d) Creating dataframes from csv and excel files.

9) **Object Oriented Programming: basic**

a) Write a Python class named Person with attributes name, age, weight (kgs), height (ft) and takes them through the constructor and exposes a method get_bmi_result() which returns one of "underweight", "healthy", "obese"
b) Write a python program to demonstrate various kinds of inheritance.

10) **Object Oriented Programming: advanced**
a) Write a python program to demonstrate operator overloading.
b) Write a python program to create abstract classes and abstract methods.

11) **Python Collections:**
a) Write a Python program to show different ways to create Counter.
b) Write a Python program to demonstrate working of OrderedDict.
c) Write a Python program to demonstrate working of defaultdict
d) Write a python program to demonstrate working of ChainMap

12) **Python collections:**
a) Write a Python program to demonstrate the working of namedtuple() and _make(), _asdict().
b) Write a Python program to demonstrate the working of deque.

13) **Regular Expressions**
a) Given an input file which contains a list of names and phone numbers separated by spaces in the following format:
i) Phone Number contains a 3- or 2-digit area code and a hyphen followed by an 8-digit number.
ii) Find all names having phone numbers with a 3-digit area code using regular expressions.
c) Write a python program to check the validity of a password given by the user. The password should satisfy the following criteria:
i) Contain at least 1 letter between a and z
ii) Contain at least 1 number between 0 and 9
iii) Contain at least 1 letter between A and Z
iv) Contain at least 1 character from $, #, @
v) Minimum length of password: 6
vi) Maximum length of password: 12
d) Write a Python program to validate mobile number.

14) Write a Python program to print checkerboard pattern of nxn using numpy

15) Write a Python program to demonstrate working of OS Module.

16) Write a Python program to demonstrate working of Calendar Module.

17) Write a Python program using pandas that finds Missing Data and replace missing **data.**