BANGALORE UNIVERSITY

ಬ'ೆಂಗಳೂರು ವಿಶ್ವವಿದ್ಯಾಲಯ

Jnanabharathi, Bengaluru, Karnataka – 560 056

SCHEME AND SYLLABUS

For the Course

BACHELOR OF COMPUTER APPLICATIONS

(Artificial Intelligence & Machine Learning/ Data Science/ Full Stack Development)

State Education Policy 2024 SCHEME

Academic Year 2024- 2025 and onwards

Board Members

Sl. No.	Name	Designation
1.	Dr. Muralidhara B. L Coordinator BCA & Senior Professor, Dept. of Computer Science and Applications, Bangalore University, Bangalore -560056.	Chairperson
2.	Dr. Somashekara M.T Associate Professor, Dept. of Computer Science and Applications, Bangalore University, Bangalore -560056.	Member
3.	Dr. Suresh R Assistant Professor, Dept. of Statistics, Bangalore University,Bangalore- 560056.	Member
4.	Dr. Kumbinarasaiah S, Assistant Professor, Dept. of Mathematics, Bangalore University,Bangalore-560056.	Member
5.	Mr. Chitty Babu G, Consultant, 112, 10th Cross, 29th Main, JP Nagar I Phase, Bangalore - 560078	Member
6.	Dr. Nazura Javed Kutty, Associate Professor, St. Francis College, 3rd Block, 8th Main, Koramangala 560 034	Member
7.	Ms. Radhika E K, Academic Co-ordinator & HOD, Computer Science dept, Sindhi College, #33/2B, Hebbal, Kempapura, Bengaluru-560024	Member
8.	Dr. C.Umarani, Associate professor, Christ Academy Institute for Advanced studies, Begur Koppa Road, Hullahalli, Sakalavara, Bangalore, 560083	Member
9.	Mr. Hemanth Uppala Assistant Professor& Vice Principal, Dayananda Sagar College of Arts, Science and Commerce, Shavige Malkeswara Hills, Kumaraswamy Layout, Bengaluru 560011.	Member
10.	Mr. Dadavali S.P Assistant Professor, Government First Grade College, Kengeri Satellite Town , Kengeri, Bengaluru 560060.	Member
11.	Ms. Nazia Hassan, Department of Computer Science, Government First Grade College, Vijayanagar-560040	Special Invitee

Course Matrix

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
1	24BCA11	Discrete Structure	03	80	20	100	3
	24BCA12	Problem Solving Technique	03	80	20	100	3
	24BCA13	Computer Architecture	03	80	20	100	3
	24BCA12P	Problem Solving Technique Lab	04	40	10	50	2
	24BCA13P	Computer Architecture Lab	04	40	10	50	2
	24BCA1P	Office Automation Tools	04	40	10	50	2
	24BCAL11	Language L1	04	80	20	100	3
	24BCAL12	Language L2	04	80	20	100	3
	24BCACC1	Environmental Studies	02	40	10	50	2
2	24BCA21	Data Structure	03	80	20	100	3
	24BCA22	Object Oriented Programming Using JAVA	03	80	20	100	3
	24BCA23	Operating Systems	03	80	20	100	5
	24BCA21P	Data Structure Lab	04	40	10	50	2
	24BCA22P	Object Oriented Programming Lab	04	40	10	50	2
	24BCA23P	LINUX and Shell Programming Lab	04	40	10	50	2
	24BCAL21	Language L1	04	80	20	100	3
	24BCAL22	Language L2	04	80	20	100	3
	24BCASEC1	Computer Applications	02	40	10	50	2

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
3	24BCA31	Database Management System	03	80	20	100	3
	24BCA32	Probability and Statistics	03	80	20	100	3
	24BCA33	Artificial Intelligence	04	80	20	100	4
	24BCA31P	Database Management System Lab	04	40	10	50	2
	24BCA32P	Artificial Intelligence Lab using Python	04	40	10	50	2
	24BCAE1	Elective: I Feature Engineering (AI &ML) Basics of Data Analytics using Spreadsheet (Data Science) Web Programming-I (Full Stack Development)	02	40	10	50	2
	24BCAL31	Language L1	04	80	20	100	3
	24BCAL32	Language L2	04	80	20	100	3
	24BCACC2	The Constitution of India	02	40	10	50	2
4	24BCA41	Computer Networks	03	80	20	100	3
	24BCA42	Design and Analysis of Algorithms	03	80	20	100	3
	24BCA43	Software Engineering	04	80	20	100	4
	24BCA41P	Computer Networks Lab	04	40	10	50	2
	24BCA42P	Design and Analysis of Algorithms Lab	04	40	10	50	2
	24BCAE2	Elective: II Introduction to ML(AI &ML) Data Visualization (Data Science) Web Programming-II (Full Stack Development)	02	40	10	50	2
	24BCAL41	Language L1	04	80	20	100	3
	24BCAL42	Language L2	04	80	20	100	3
	24BCASEC2	Computer Assembly and Repair Lab	04	40	10	50	2

	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(AI & ML)							
5	24BCA51	ML & Neural Network	03	80	20	100	3
	24BCA52	Digital Image Processing	03	80	20	100	3
	24BCA53	Natural Language Processing	05	80	20	100	5
	24BCA51P	ML & Neural Network Lab	04	40	10	50	2
	24BCA52P	Digital Image Processing Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (AI & ML)							
6	24BCA61	Deep Learning for Computer Vision	04	80	20	100	5
	24BCA62	Predictive Analysis	04	80	20	100	5
	24BCA63	Project Work	10	80	20	100	5
	24BCASEC3	Soft Skills	02	40	10	50	2
Semester – V (Data Science)							
5	24BCA51	Introduction to Data Science	03	80	20	100	3
	24BCA52	Time Series Analysis	03	80	20	100	3
	24BCA53	Machine Learning	03	80	20	100	3
	24BCA51P	Data Science Lab	04	40	10	50	2
	24BCA52P	Time Series Analysis Lab	04	40	10	50	2
	24BCA53P	Machine Learning Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (Data Science)							
6	24BCA61	Big Data Analytics	05	80	20	100	5
	24BCA62	Exploratory Data Analysis	05	80	20	100	5
	24BCA63	Project Work	10	80	20	100	5
	24BCASEC3	Soft Skills	02	40	10	50	2

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(Full Stack Development)							
5	24BCA51	Frontend Design	03	80	20	100	3
	24BCA52	Backend Development	03	80	20	100	3
	24BCA53	Cloud computing	05	80	20	100	5
	24BCA51P	Frontend Design Lab	04	40	10	50	2
	24BCA52P	Backend Development Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (Full Stack Development)							
6	24BCA61	Software Project Management	04	80	20	100	5
	24BCA62	Mobile Application Development	04	80	20	100	5
	24BCA63	Project Work	10	80	20	100	5
	24BCASEC3	Soft Skills	02	40	10	50	2

Program Outcome

PO1	Computational Knowledge	Acquire in-depth computational and mathematical knowledge with an ability to abstract and conceptualise from defined problems and requirements.
PO2	Dynamic Problem-Solving Skill	Identify, formulate, and exhibit strong analytical and dynamic problem-solving skills to address evolving computational challenges.
PO3	Innovative System Analysis and Design/ Development	Design and evaluate solutions for complex problems in Data Science, AI & ML, and Full Stack Development, considering societal, cultural, and environmental factors.
PO4	Investigate complex computing problem	Conduct literature surveys, analyze information, and design experiments using appropriate research methods to derive valid conclusions in relevant domains.
PO5	Use of modern tools/ Adaptive programming proficiency	Select, adapt, and apply modern IT tools and programming languages effectively in Data Science, AI & ML, and Full Stack Development to solve diverse computing challenges.
PO6	Knowledge Optimization	Modify algorithms or software systems to improve efficiency or resource utilization.
PO7	Life Long Continuous learning and Technology Adaptability	Pursue lifelong learning to stay updated with emerging technologies in Data Science, AI & ML, and Full Stack Development for sustained employability.
PO8	Soft skills and collaborative teamwork	Communicate effectively, enhance interpersonal skills, and collaborate in multidisciplinary teams essential for success in professional environments.
PO9	Cyber Security Proficiency	Understand cyber threats, develop secure software, and protect sensitive data in Data Science, AI & ML, and Full Stack Development applications.
PO10	Ethical and Professional Conduct	Adhere to ethical standards and professional practices in Data Science, AI & ML, and Full Stack Development roles and responsibilities.
PO11	Employability	Identify market trends, upgrade skills accordingly, and enhance employability in Data Science, AI & ML, and Full Stack Development careers.
PO12	Innovation and Entrepreneurship	Identify opportunities, innovate, and create value through Data Science, AI & ML, and Full Stack Development projects for personal growth and societal impact.

SEMESTER – I

DISCRETE STRUCTURES

Course Outcome

CO1 Apply Set theory, Relations, Functions and Mathematical Induction to solve problems

CO2 Getting familiar with counting techniques and Mathematical Logic, and apply the same to address the real-life problems

CO3 Learn different operations of a Matrix and apply them to solve real-life problems.

CO4 Apply graph theory concepts to solve the problems of real world.

UNIT – I 11 Hours

Set Theory: Fundamentals of Set theory, Set Operations, Laws of Set Theory, Counting and Venn Diagrams, Cartesian Product, Relations, Types of Relations, Functions, Types of Functions, Function Composition, Inverse Functions. Mathematical Induction.

UNIT – II 11 Hours

Logic and Counting: Fundamentals of Logic, Propositional Logic, Logical Connectives and Truth Tables, Logic Equivalence, Tautology and Contradiction. Basics of counting, Counting Principles, Pigeonhole Principle, Permutation, Combinations.

UNIT - III 11 Hours

Matrices: Basics of Matrix, Types of Matrices, Operations on Matrices, Inverse of a matrix, Solution for system of linear equations, Determinant, Properties of Determinant, Cramer's Rule, Introduction to Eigen Values and Eigen Vectors.

UNIT - IV 12 Hours

Graph Theory: Graphs: Introduction, Representing Graphs, Operations on graphs, Directed Graphs Graph Isomorphism, Paths, Cycles, Euler Graph, Hamilton Graph, Planar Graphs.
Trees: Introduction, Applications of Trees, Spanning Trees, Minimum Spanning Trees, Prim's and Kruskul's Algorithms.

Text Book

1 Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education, 2006.

2 Richard Bronson, Schaum's Outline of Matrix Operations, McGraw-Hill publications, 2nd Edition,

Reference Books

1 Gregory Hartman, Fundamentals of Matrix Algebra, Third Edition,2011.

2 Gary Haggard, John Schlipf and Sue Whitesides: Discrete Mathematics for Computer Science, (with Student Solutions Manual CD-ROM), Cengage Learning; 1st edition, 2005.

Course Articulation Matrix: Mapping of Course Outcomes (COs)with Program Outcomes(POs1-12)

Course Outcome(COs)	**Program Outcomes(POs)**											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	2				1						
CO2	2	3		1			1					
CO3	2	1						1				
CO4	1	1	1									

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

PROBLEM-SOLVING TECHNIQUE

Course Outcomes

Upon successful completion of the course, the student will be able:

CO1 To understand algorithmic strategies for enhancing problem-solving proficiency

CO2 Demonstrate problem-solving tools and techniques using C.

CO3 To analyze the given problems and use appropriate algorithms.

CO4 To implement sorting and searching techniques to develop programs.

UNIT –1 12 Hours

Introduction: The Role of Algorithms in computing, Algorithms as a technology, analyzing algorithms, Designing algorithms, Flow charts. Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of Fibonacci sequence, Reversing the digits of an integer, Character to number conversion.

11 Hours

UNIT-II

C Programming: Getting Started, Variables, Operators and Arithmetic expressions. Input and Output: Standard input and output, formatted input and output. Selection statements: Statements and Blocks, If, If-else, if-else-if ladder, nested if, switch. Control Structure: while loop, for loop, do-while loop, break and continue, go to and labels. Pointers and Arrays: Pointers and address, Pointers and function arguments, One-Dimensional array, Two-Dimensional array, Structures and Union, Command line arguments.

UNIT - III 11 Hours

Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factor of an integer, raising a number to a large power. Array Techniques: Array order reversal, Array counting, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, finding the k^{th} smallest element, and multiplication of two matrices.

UNIT - IV 11 Hours

Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: Linear Search, Binary search. Text processing and Pattern searching: Text line length adjustment, keyboard searching in text, text line editing, linear pattern searching.

Text Book

1 R. G. Dromey, "How to Solve it by Computer", Person Education India, 2008.

2 Brain M. Kernighan and Dennis M. Ritchie, " The C Programming Language", 2nd edition, Princeton Hall Software Series, 2012.

3 Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2008.

Reference Books

1 E. Balaguruswamy, "Programming In ANSI C", 4th edition, TMH Publications, 2007

2 Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.

3 Donald E. Knuth, The Art of Computer Programming", Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome(COs)	**Program Outcomes(POs)**											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	3	1	1	1	1	2	1	1	1	2	1
CO2	3	3	3	2	1	1	1	1	1	1	2	1
CO3	3	3	1	1	1	1	1	1	1	1	2	1
CO4	3	3	2	2	1	1	1	1	1	1	2	1

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

COMPUTER ARCHITECTURE

Course Outcome

CO1 Understand various arithmetic and logical operations on different types of numbers to design an arithmetic and logic unit.

CO2 Demonstrate Design and implement sequential logic circuits using ICs

CO3 Analyze the basics of computer organization and its design and the basic processing unit

CO4 Implement the instruction sets and to develop assembly language programming skills.

UNIT-1 [12Hours]

Number Systems: Decimal, Binary, Hexadecimal, Octal Number System Conversions, Binary Arithmetic, Complements- r's complement, (r-1)'s complement, Addition and subtraction of BCD, Octal Arithmetic, Hexadecimal Arithmetic, Binary Codes, Decimal Codes, Error detecting and correcting codes, ASCII, EBCDIC, UNICODE, Digital Logic Circuits: Digital Computers, Logic Gates, Universal Gates, Boolean algebra, Map Simplification.

UNIT-2 [11Hours]

Combinational Circuits- Half Adder and Full Adder, Flip-Flops- SR Flip- Flop, D Flip-Flop, J-K Flip-Flop, T Flip-Flop, Sequential Circuits- Flip-Flop input equations, State Table, State Diagram and problems. Digital Components: Integrated Circuits, Decoders-3-to-8-line decoder, NAND gate Decoder, Octal to Binary Encoder, Multiplexers- 4-to-1 line Multiplexer, Registers- 4 bit register with parallel load, Shift Registers- Bidirectional shift register with parallel load, Binary Counters-4-bit synchronous binary counter.

UNIT-3 [11Hours]

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator logic. Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC), CISC Vs RISC.

UNIT-4 [11Hours]

Introduction to 8085 Assembly language programming: Architecture of 8085, Pin Configuration, The 8085-programming model, Instruction classification, Instruction, data formats, and storage. How to write assemble and execute a simple program, overview of 8085 instruction set. Introduction to 8085 Instructions: Instruction classification of 8085 based on word length and functions, Data Transfer operations, Arithmetic operations, Logic Operations, Brach operations, Writing Assembly language programs, Addressing modes of 8085.

Text Book

1 M. Morris Mano- "Computer System Architecture", 3rd Edition Pearson India, 2019.

2 Ramesh Gaonkar- "Microprocessor Architecture, Programming and Applications with the 8085", 5th Edition, Penram International Publishing (India) Private Limited,2007.

3 Andrew S. Tanenbaum, Todd Austin –"Structured Computer Organization", PHI Pearson 6th, Edition,2013.

Reference Books

1 William Stallings- "Computer Organization and Architecture", Pearson/PHI, 6th Edition,2007.

2 Andrew S. Tanenbaum-" Structured Computer Organization", Pearson 4th Edition,1998.

3 M.V .Subramanyam, "Switching Theory and Logic Design", Laxmi Publications (P) Ltd,2011.

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome (COs)	Program Outcomes(POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	3	1	1	1	1	1	1	1	1	2	1
CO2	3	2	2	3	1	1	1	1	1	1	2	1
CO3	3	2	2	1	2	2	1	1	1	1	1	1
CO4	3	2	2	1	2	2	1	1	1	1	2	1

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

PROBLEM-SOLVING TECHNIQUE LAB

Write, and execute C Program for the following:

1. To read the radius of the circle and to find area and circumference.
2. To read the numbers and find the biggest of three.
3. To check whether the number is prime or not.
4. To find the root of quadratic equation.
5. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
6. To read the numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
7. To read percentage of marks and to display appropriate message. If a percentage is 70 and above- Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail.(Demonstrate of if-else ladder)
8. To simulate a simple calculator with addition, subtraction, multiplication, division and it should display the error message for division of zero using switch case.
9. To read marks scored by n students and find the average of mark (Demonstration of single dimensional array)
10. To remove duplicate elements in a single dimensional array.
11. To find the factorial of a number.
12. To generate Fibonacci series.
13. To demonstrate string functions. (String Length, String Copy, String Concatenate, String Comparison)
14. To find the length of the string without using built-in function.

15. To read, display and add two n x m matrices using function.
16. To read a string and to find the number of alphabets, digits, vowels, consonants, space and special characters.
17. To swap two numbers using pointers.
18. To demonstrate student structure to read & display records of n students.
19. To demonstrate the difference between structure and union for the following Student name (String), Student roll no(integer), Student mark(float)
20. To design the following pattern using nested for loop:

```
        *

      *   *

    *   *   *

  *   *   *   *

*   *   *   *   *
```

COMPUTER ARCHITECTURE LAB

1) Write an 8085 Program to swap two 8-bit numbers.
2) a. Write a Program to find the largest of two numbers
   b. Write an 8085 Program to find the smallest of two numbers
3) Write an 8085 Program to find whether an 8-bit number is positive, negative or zero. If positive display EE, if negative display FF, if zero display DD.
4) Write an 8085 Program to check whether 4^{th} bit of a number is zero or one. If 4^{th} bit is 1 display FF, if 4^{th} bit is 0 display DD.
5) Write an 8085 Program to calculate the sum of first ten natural numbers.
6) Write an assembly language program in 8085 microprocessors to find the sum of digits of an 8-bit number.
7) Write an 8085 Program to find the reverse of an 8-bit number
8) Write an 8085 Program to check whether 1-byte number is a palindrome or not. If it is a palindrome display FF otherwise display DD.
9) Write an 8085 Program to check whether a number is ODD or EVEN. If Even no. display DD, if odd no. display FF.
10) Write an 8085 program to count a number of ones in the given 8-bit number.
11) Write an 8085 program to find Addition & Subtraction of two 8 –bit HEX numbers.
12) Write an 8085 program to find Addition of two 16 –bit numbers.
13) Write an 8085 program to find Subtraction of two 16 –bit numbers.

14) Write an 8085 program for Swapping of two 16-bit numbers.
15) Write an 8085 program to implement 2 out of 5 codes
16) Write an 8085 program to generate Fibonacci series
17) Write an 8085 program to find the first ten terms of odd and even numbers.
18) Write an 8085 program to find 4-Digit BCD addition.
19) Write an 8085 program to find Multiplication of 2-digit BCD numbers.
20) Write an 8085 program to find division of two 8-bit numbers

SEMESTER –II

DATA STRUCTURES

Course Outcome

CO1 Understand basic concepts of data structures.
CO2 Analyzing and exploring various ways of storing data using Array and Linked list.
CO3 Demonstrate stack and queue data structures and their applications
CO4 Analyze and implement various nonlinear data structures.

UNIT I 11 Hours

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade-off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. Introduction to Strings, Storing String, Character Data Types, String Operations, word processing, Introduction to pattern matching algorithms.

UNIT II 11 Hours

Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, multi-dimensional arrays, Matrices and Sparse matrices, searching and sorting techniques using array. Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.

UNIT III 11 Hours

Stacks: Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Postfix expression, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack. Queues: Definition, Array representation of queue, Linked list representation of queues. Types of queue: Simple queue, Circular queue, Double-ended queue, Priority queue, Operations on Queues, Applications of queues.

UNIT IV 12 Hours

Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Red Black Tree: Insertion and Deletion, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Background, Computer Representation, Graph Traversal. Hashing: Hash Table ADT, understanding Hashing, Components of Hashing, Hash Table, Hash Function, Hashing Techniques, collisions, collision resolution techniques.

Text Book

1 Seymour Lipschutz, “Data Structures with C”, Schaum’s Outlines, Tata Mc Graw Hill, 2011.

2 Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla,“Data Structures and Program Design using C”, Pearson Education, 2009

Reference Books	
1	Mark Allen Weiss,“ Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education,2013

2	Forouzan,"A Structured Programming Approach using C",2nd Edition, Cengage LearningIndia,2008.

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs12)

Course Outcome (COs)	Program Outcomes (POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	4	3	4	4	4	3	2	3	1	4	4
CO2	5	5	4	4	4	4	3	2	3	1	4	4
CO3	5	5	4	4	4	4	3	2	3	1	4	5
CO4	5	5	4	4	4	4	3	2	2	1	4	4

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

OBJECT-ORIENTED PROGRAMMING USING JAVA

Course Outcome

CO1 - Understand object-oriented programming concepts

CO2 - Demonstrate the important feature of Object-oriented programming

CO3 - Examine event handling, String handling, and exception handling concepts

CO4 - Implement concepts to solve real-world problems

UNIT-1 [12 Hours]

Introduction: Basics of object-oriented programming, comparison of procedure-oriented and object-oriented programming paradigms; Difference between C and Java Programming languages; Features of Java; Objects and classes in Java, Structure of a Java program; Data Types, variables and operators in java; Control structures- Branching and looping; Methods & Constructors in java; Java Development Kit (JDK); Built-in classes in Java; Math, Character, String, String Buffer and Scanner; Wrapper classes; The abstract, static and final classes; Casting objects; The instance of operator; Usage of this keyword; Arrays in Java.

[11 Hours]

UNIT-2

Inheritance: Super and subclasses; visibility modifiers; Types of Inheritance- single, multilevel, hierarchical and hybrid inheritance; the interface concept in Java, Polymorphism: Compile time and run time polymorphisms – Method overloading and method overriding. Package: Types of packages; the util, awt and swing packages; Creating and importing user-defined packages. I/O programming: Standard I/O streams in Java; Types of streams – Based on the type of Operations and the type of file.

[11 Hours]

UNIT-3

Event handling: Major events in Java; Two Event Handling mechanisms- Event classes and Event Listener Interfaces; Mouse and keyboard events; GUI: Panels; Frames; Layout managers – Flow, border and grid layouts; Buttons; Checkboxes; Radio buttons; Labels; Text fields; Text areas; Combo boxes; Scroll bars; Sliders; Menu, Dialog boxes. Applet programming: Comparison of applets and applications; Applet life cycle; Developing and running applets. String handling: String construction, string length, special string operations, character extraction, string comparison, modifying string and string buffers.

UNIT-4 [11 Hours]

Exception handling: Types of Java exception – checked and unchecked exceptions; Usage of try-catch-finally blocks. Multithreading: comparison of multithreading and multitasking; Life cycle of a thread; two ways of creating thread – by extending the Thread class and by implementing the Runnable Interface, Thread synchronization. Advanced concepts: Collections in Java; Introduction to JavaBeans and Java security manager, Importance of generic programming in java with examples.

Text Book

1 E. Balagurusamy, Programming with JAVA, McGraw Hill, New Delhi, 2007

2 Herbert Schildt, Java A Beginner's Guide – Create, Compile, and Run Java Programs Today, Sixth Edition, Oracle Press, 2014.

Reference Books

1 Ken Arnold, James Gosling, "The Java Programming Language, Fourth Edition, Addison Wisely, 2005

2 R Herbert Schildt, 'The Complete Reference Java, 7th Edition, McGraw Hill, 2007

Course Articulation Matrix: Mapping of Course Outcomes (COs)with Program Outcomes (POs1-12)

Course Outcome (COs)	**Program Outcomes (POs)**											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	4	3	2	2	3	2	3	2	2	2	4	2
CO2	4	3	3	3	4	4	3	2	3	3	4	3
CO3	4	4	3	3	4	3	3	2	3	3	4	4
CO4	4	4	4	3	4	4	3	2	3	3	4	4

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per EP guidelines are compulsory	

OPERATING SYSTEMS

Course Outcomes

CO1 To analyze the memory management and its allocation policies

CO2 To understand synchronization and deadlock conditions and their possible solutions

CO3 To discuss the storage management policies with respect to different storage

CO4 To evaluate the concept of the operating system with respect to Linux architecture and commands

UNIT –1 [11 Hours]

Introduction: Computer System Organization, Architecture, Structure, Operations, Process Management, Memory Management, Storage Management. Operating System Structures: Services, System Calls, Types, Operating System Structure, System Boot. Processes: Process Concept, Scheduling, Operations, Inter-process Communication. Multithreaded Programming: Multithreading Models

UNIT-II [12 Hours]

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples. Process Scheduling: Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-time CPU Scheduling. Deadlocks: System model, Characterization, Methods for handling deadlocks, Deadlock Prevention, Avoidance, Detection and Recovery from deadlock.

UNIT – III [11 Hours]

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Demand Paging; Copy-on-Write, Page Replacement; Allocation of Frames, File System: File Concept, Access Methods, Directory and Disk Structure, Protection. File-system Implementation: Structure, File-System and Directory Implementation, Allocation Methods, Free Space Management. Mass-Storage Structure: Overview, Disk Scheduling, Disk Management.

UNIT – IV [11 Hours]

Introduction to Linux Programming: Linux system Architecture, Linux Command format, Linux Internal and External Commands, Directory Commands, File related commands, Disk related commands, General Utilities

Text Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; “Operating Systems Concepts”, 9th Edition, 2016 India, Wiley.
2. William Stallings, "Operating Systems and Design Principles", Pearson, 5th Edition,2018

Reference Books

1. D M Dhamdhere : Operating Systems - A concept Based Approach, 3rd Edition, Tata McGraw

- Hill, 2017.
2. Sumitabha Das: "UNIX Concepts and Applications", 4th Edition, Tata McGraw Hill, 2017.
3. M G Venkateshmurthy, Introduction to UNlX and Shell Programming", Pearson Edition, 2005.

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome(COs)	Program Outcomes(POs)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	4	3		4		2					4	2
CO2	4	2		3		2		3			2	2
CO3	3	3		4		2					2	2
CO4	3	2		2		2					4	2

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per EP guidelines are compulsory	

DATA STRUCTURES LAB

NOTE: For all the programs write the output, flowchart and number of basic operations performed.

1. Write a program to search for an element in an array using binary and linear search.
2. Write a program to sort list of n numbers using Bubble Sort algorithms.
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {45, 34, 10, 63,3} into linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to simulate the working of Circular queue using an array.
7. Write a program to insert the elements {61,16,8,27} into ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program for Tower of Honoi problem using recursion.
9. Write recursive program to find GCD of 3 numbers.
10. Write a program to demonstrate working of stack using linked list.
11. Write a program to convert an infix expression x^y/(5*z)+2 to its postfix expression
12. Write a program to evaluate a postfix expression 5 3+8 2 - *.
13. Write a program to create a binary tree with the elements {18,15,40,50,30,17,41} after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation.
14. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
15. Write a program to Sort the following elements using heap sort {9.16,32,8,4,1,5,8,0}.
16. Given S1={“Flowers”} ; S2={“are beautiful”} I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring “low” from S1 IV. Find “are” in S2 and replace it with “is” .
17. Write a program to implement adjacency matrix of a graph.
18. Write a program to insert/retrieve an entry into hash/ from a hash table with open addressing using linear probing.

OBJECT ORIENTED PROGRAMMING LAB

1. Java program to display “Hello World” and display the size of all the data types.
2. Java program to implement the usage of static, local and global variables.
3. Java program to implement string operations string length, string concatenate, substring
4. Java program to find the maximum of three numbers
5. Java program to check whether the number is odd or even.
6. Java program to implement default and parameterized constructors.
7. Java program to implement an array of objects.
8. Java program to implement Single Inheritance
9. Java program to implement Multiple Inheritance using Interface
10. Java program to implement an applet
11. Java program to demonstrate a division by zero exception
12. Java program to add two integers and two float numbers. When no arguments are supplied give a default value to calculate the sum. Use method overloading.
13. Java program that demonstrates run-time polymorphism.
14. Java program to catch negative array size Exception. This exception is caused when the array is initialized to negative values.
15. Java program to handle null pointer exception and use the “finally” method to display a message to the user.
16. Java program to import user-defined packages
17. Java program to check whether a number is palindrome or not
18. Java program to find the factorial of a list of numbers reading input as command line argument.
19. Java program to display all prime numbers between two limits.
20. Java program to create a thread using Runnable Interface.

LINUX AND SHELL PROGRAMMING LAB

1. Write a shell script to print all the prime numbers between M to N (M<N).
2. Write a shell script to reverse a given number and check whether it is a palindrome.
3. Write a shell script to find the sum of digits of a given number using loops and without using loops.
4. Write a shell script to implement 10 Linux commands using case.

5. Write a Shell script that displays a list of all the files in the current directory to which the user
6. has read, write and execute permissions?
7. Write a shell script to copy a file within the current directory
8. Write a shell script to copy file between two directories
9. Write a Shell script to create two data files and compare them to display unique and common entries.
10. Write a shell script to count the number of vowels in a string.
11. Write a shell script to convert uppercase characters to lowercase and vice versa.
12. Write a shell script to accept a word and perform pattern matching in a given file.
13. Write a shell script to find the factorial of a number
14. Write a Menu-driven program to demonstrate the zombie process and orphan process.

SEMESTER –III

24BCA31	Database Management Systems	45 Hours

Course Objectives

CO1: Understanding Core Concepts of DBMS
CO2: Proficiency in Database Design and SQL
CO3: Application of Advanced Database Techniques

Prerequisite: Basic knowledge of Set Theory, Operating systems, Discrete mathematics, Data structures, and Programming.

Course Content:

UNIT I: [11 Hours]

Introduction to Databases: Definition of Data, Database, and DBMS, Overview of Database Applications, Advantages and Disadvantages of DBMS, Roles of Database Users and Administrators

Data Models: Introduction to Data Models, Types of Data Models (Hierarchical, Network, Relational, Object-oriented), Importance of Data Models in DBMS

Database Design: Keys: Primary Key, Candidate Key, Super Key, Foreign Key, Composite Key, Alternate Key, Unique Key, Surrogate Key, Constraints in a table: Primary Key, Foreign Key, Unique Key, NOT NULL, CHECK, Entity-Relationship (ER) Model, Entities and Entity Sets, Attributes and Relationships, ER Diagrams, Key Constraints and Weak Entity Sets, Extended ER Features, Introduction to the Relational Model and Relational Schema

UNIT II: [12 Hours]

Relational Algebra and Calculus: Introduction to Relational Algebra, Operations: Selection, Projection, Set Operations, Join Operations, Division,Tuple and Domain Relational Calculus

Structured Query Language (SQL): SQL Basics: DDL and DML, Aggregate Functions (Min(), Max(), Sum(), Avg(), Count()), Logical operators (AND, OR, NOT), Predicates (Like, Between, Alias, Distinct), Clauses(Group By, Having, Order by, top/limit), Inner Join, Natural Join, Full Outer Join, Left Outer Join, Right outer Join, Equi Join

Normalization and Database Design: Functional Dependencies: Armstrong's Axioms, Definition, Properties (Reflexivity, Augmentation, Transitivity), Types (Trivial, Non-Trivial, Partial and Full Functional Dependency), Closure of Functional Dependencies, Normal Forms (1NF, 2NF, 3NF, BCNF), Denormalization.

UNIT III: [11 Hours]

Transaction Management:ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking,

Transaction Support in SQL, Introduction to Crash Recovery, 2PL, Serializability, and Recoverability, Introduction to Lock Management, Dealing with Deadlocks.

Database Storage and Indexing: Data on External Storage, File Organizations and Indexing, Index Data Structures, Comparison of File Organizations, Indexes and Performance Tuning, Guidelines for Index Selection, Basic Examples of Index Selection

UNIT IV: [11 Hours]

NoSQL Databases and Big Data: Introduction to NoSQL, Data Models: Document, Key value, Column family, Graph. Uses and Features of NO/SQL document databases. CAP theorem, BASE vs ACID, CRUD operations, MongoDB operators, Overview of Big Data Technologies: Hadoop, MongoDB, Cassandra.

Text Books

1. Elmasri and Navathe, "Fundamentals of Database Systems", 7th Edition Addison-Wesley, 2016.
2. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", third edition, McGraw – Hill, 2018
3. Benjamin Rosenzweig, Elena Rakhimov, "Oracle PL/SQL by Example", fifth edition, Prentice Hall, 2015
4. Brad Dayley, "NoSQL with MongoDB in 24 Hours", 1st edition, Sams Publishing, 2024

Reference Books

1. Korth, Silbertz, Sudarshan," Database System Concepts", Seventh Edition, McGraw - Hill.(2019)
2. R.P. Mahapatra, Govind Verma, "Database Management Systems", Khanna Publishing House, 2025.

Web Resources

1. https://oracle-base.com/articles
2. https://forums.oracle.com/ords/apexds/domain/dev- community/category/sql_and_pl_sql
3. https://asktom.oracle.com/ords/f?p=100:1:0

24BCA32	Probability and Statistics	45 Hours

Course Objectives

CO1: This course aims to make the students trained to handle randomness scientifically using theory of probability.

CO2: This course intends to make the students able to represent the statistical data in a systematic way and analyze it to draw meaningful information from them.

CO3: Through plentiful examples and exercises, this course provides the students scope to apply probabilistic and statistical techniques to deal with the real-life problems.

Course Content:

UNIT I: [11 Hours]

Basic concepts of Statistics, qualitative and quantitative data, classification of data, construction of frequency distribution, diagrammatic representation of data.

Measures of Central Tendency: Arithmetic mean, median and mode—their properties **Measures of Dispersion**: Range, mean deviation, quartile deviation, variance and standard deviation.

UNIT II: [12 Hours]

Correlation: Definition, scatter diagram, types of correlation, measures—Karl Pearson's correlation coefficient and Spearman's rank correlation coefficient.

Regression: Linear regression-fitting by least square method and interpretation.

UNIT III: [11 Hours]

Concepts of probability: Experiment and sample space, events and operations with events, probability of an event, basic probability rules, applications of probability rules, conditional probability.

Random Variables: Discrete and continuous random variable**,** probability distribution of a random variable**,** probability mass function, probability density function, expectation and variance of a random variable.

Standard Probability Distributions: Binomial probability distribution, Poisson probability distribution, Normal probability distribution.

UNIT IV: [11 Hours]

Sampling Distribution: Concept of Population and Sample, parameter and statistic, sampling distribution of sample mean and sample proportion.

Statistical Inference: Estimation and Hypothesis Testing (only concept).

Text Books

1. Manish Sharma, Amit Gupta, The Practice of Business Statistics, Khanna Book Publishing Company, 2010 (AICTE Recommended Textbook)
2. Das N. G., Statistical Methods, Combined Edition, Tata McGraw Hill, 2010.
3. Ross Sheldon M., Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Elsevier, 2021.
4. Miller Irwin and Miller Marylees, Mathematical Statistics with Applications, Seventh Edition, Pearson Education, 2005

Reference Books

1. Pal Nabendu and Sarkar Sahadeb, Statistics: Concepts and Applications, Second Edition, PHI, 2013
2. Montgomery Douglas and Runger George C., Applied Statistics and Probability for Engineers, Wiley, 2016.
3. Reena Garg, Engineering Mathematics, Khanna Publishing House, 2024.

Web Resources

1. https://nptel.ac.in/courses/111106112
2. https://nptel.ac.in/courses/111105041

24BCA33	Artificial Intelligence	60 Hours

Prerequisites:

Basic understanding of computer science concepts, including data structures and algorithms. Proficiency in minimum one programming language, such as Python.

Course Content:

UNIT I: Introduction to AI **[15 Hours]**

What is AI? Intelligent Agents: Agents and environment, the concept of Rationality, the nature of environment, the structure of Agents. Knowledge-Based Agents**:** Introduction to Knowledge-Based Agents, The Wumpus World as an Example World. Problem-solving: Problem-solving agents.

UNIT II: Advanced Search Techniques **[15 Hours]**

Uninformed Search: DFS, BFS, Iterative Deepening Search. Informed Search: Best First Search, A* search, AO* search. Adversarial Search & Games: Two-player zero-sum games, Minimax Search, Alpha-Beta pruning. Constraints and Constraint Satisfaction Problems (CSPs), Backtracking search for CSP. Evolutionary Search Techniques**:** Introduction to evolutionary algorithms, Genetic algorithms, Applications of evolutionary search in AI.

UNIT III: Logical Reasoning and Uncertainty **[15 Hours]**

Logic: Propositional logic, First-order predicate logic, Propositional versus first-order inference, Unification and lifting. Inference: Forward chaining, Backward chaining, Resolution, Truth maintenance systems. Introduction to Planning: Blocks World problem,

Strips; Handling Uncertainties: Non-monotonic reasoning, Probabilistic reasoning, Introduction to Fuzzy set theory.

UNIT IV: Domains and Applications of AI **[15 Hours]**

Domains in AI: Introduction to Machine Learning, Computer Vision, Robotics, Natural Language Processing, Deep Neural Networks, and their Applications. Expert Systems: The

architecture and role of expert systems include two case studies. Legal and Ethical Issues: Concerns related to AI.

Text Books:

1. M.C. Trivedi, *A Classical Approach to Artificial Intelligence*, Khanna Book Publishing Company, 2024 (AICTE Recommended Textbook).
2. Nilsson Nils J, *Artificial Intelligence: A new Synthesis*, Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN: 978-1-55-860467-4.
3. Dan W Patterson, *Introduction to Artificial Intelligence & Expert Systems*, PHI Learning 2010.
4. Rajiv Chopra, *Data Science with Artificial Intelligence, Machine Learning and Deep Learning*, Khanna Book Publishing Company, 2024.

Reference Books:

1. M.C. Trivedi, *Introduction to AI and Machine Learning*, Khanna Book Publishing Company, 2024.
2. Russell, S. and Norvig, P., "Artificial Intelligence - A Modern Approach", 3rd edition, Prentice Hall.
3. Van Hirtum, A. & Kolski, C. (2020). *Constraint Satisfaction Problems: Algorithms and Applications*. Springer
4. Rajiv Chopra, Machine Learning and Machine Intelligence, Khanna Book Publishing Company, 2024.

Course Outcomes:

CO1: Understand the characteristics of rational agents, and the environment in which they operate, and gain insights about problem-solving agents.

CO2: Gain insights about Uninformed and Heuristic search techniques and apply them to solve search applications.

CO3: Appreciate the concepts of knowledge representation using Propositional logic and Predicate calculus and apply them for inference/reasoning.

CO4: Obtain insights about Planning and handling uncertainty through probabilistic reasoning and fuzzy sets.

CO5: Obtain a basic understanding of the AI domains and their applications and examine the legal and ethical issues of AI.

24BCA31P	Database Management Systems Lab

List of Practicals:

1. Draw an ER Diagram of Registrar Office
2. Draw an ER Diagram of Hospital Management System
3. Reduce The ER diagram in question no 1 into tables
4. Reduce the ER diagram of question no 2 into tables

Consider the following Schema
Supplier(SID, Sname, branch, city, phone)
Part(PID, Pname, color, price)
Supplies(SID, PID, qty, date_supplied)

DDL Commands

5. Create the above tables
6. Add a new attribute state in supplier table
7. Remove attribute city from supplier table
8. Modify the data type of phone attribute
9. Change the name of attribute city to address
10. Change a table's name, supplier to sup
11. Use truncate to delete the contents of supplies table
12. Remove the part table from database

DML Commands

1. Insert at least 10 records in tables supplier, part and supplies
2. Show the contents in tables supplier, part and supplies
3. Find the name and city of all suppliers
4. Find the name and phoneno of all suppliers who stay in 'Delhi'
5. Find all distinct branches of suppliers
6. Delete the record of the supplier whose SID is 204001
7. Delete all records of supplier table
8. Delete all records of suppliers whose city starts with capital A.
9. Find the supplier names which have 'lk' in any position
10. Find the supplier name where 'R' is in the second position
11. Find the name of supplier whose name starts with 'V' and ends with 'A'
12. Change the city of all suppliers to 'BOMBAY'
13. Change the city of supplier 'Vandana' to 'Goa'

Queries with Constraints

1. Create the supplier table with Primary Key Constraint
2. Create supplies table with Foreign key Constraint
3. Create a part table with UNIQUE Constraint
4. Create supplier Table with Check Constraints
5. Create Supplier table with Default

Constraint Queries on TCL

1. Create Savepoints
2. Rollback to SavePoints
3. Use Commit to save on

Aggregate Functions:

1. Find the minimum, maximum, average and sum of costs of parts
2. Count the total number of parts present
3. Retrieve the average cost of all parts supplied by 'Mike'

Queries on GROUP BY, HAVING AND ORDER BY Clauses

1. Display total price of parts of each color
2. Find the branch and the number of suppliers in that branch for branches which have more than 2 suppliers
3. Find all parts sorted by pname in ascending order and cost in descending order
4. Find the branch and the number of suppliers in that branch

Queries on Analytical, Hierarchical, Recursive nature.

1. Find out the 5^{th} highest earning employee details.
2. Which department has the highest number of employees with a salary above $80,000, and what percentage of employees in that department have a salary above $80,000
3. Retrieve employee table details using the hierarchy query and display that hierarchy path starting from the top level indicating if it is a leaf and there exists a cycle.
4. What is the average salary for employees in the top 2 departments with the highest average salary, and what is the hierarchy of departments and sub-departments for these top 2 departments?
5. Use recursion to retrieve the employee table and display the result in breadth first and depth first order.
6. Write a recursive query to show the equivalent of level, connect_by_root and connect_by_path
7. Use recursion to retrieve the employee table and display the result in depth first order showing id, parent_id, level, root_id, path and leaf.

Queries on Operators

1. Find the pname, phoneno and cost of parts which have cost equal to or greater than 200 and less than or equal to 600.
2. Find the sname , SID and branch of suppliers who are in 'local' branch or 'global' branch
3. Find the pname, phoneno and cost of parts for which cost is between 200 and 600
4. Find the pname and color of parts, which has the word 'NET' anywhere in its pname.
5. Find the PID and pname of parts with pname either 'NUT' or 'BOLT'
6. List the suppliers who supplied parts on '1st may2000', '12 JAN 2021' ,'17 dec 2000' , '10 Jan 2021'
7. Find all the distinct costs of parts

Join Operators

1. Perform Inner join on two tables
2. Perform Natural Join on two tables
3. Perform Left Outer Join on tables
4. Perform Right Outer join on tables
5. Perform Full Outer Join on tables

Set Theory Operators

1. Show the use of UNION operator with union compatibility
2. Show the use of intersect operator with union compatibility
3. Show the use of minus operator with union compatibility
4. Find the cartesian product of two tables

Queries on Set Theory Operators

1. List all parts except 'NUT' and 'BOLT' in ascending order of costs
2. display all parts that have not been supplied so far
3. To display the supplier names who have supplied 'green' part with cost 500 Rupees AND 'red' part with cost 400 Rupees.
4. To display the supplier names who have supplied 'green' part with cost 500 Rupees OR 'red' part with cost 400 Rupees.
5. To Display the name of suppliers who have supplied all parts that are 'red' in color.

PL/SQL Programs

1. Write a PL/SQL Code to add two numbers
2. Write a PL/SQL code for Fibonacci series
3. Write a PL/SQL Code for greatest of 3 numbers
4. Write a PL/SQL code for area and circumference of a circle

MongoDB Queries

1. Create a collection and insert documents into it using insertOne() and insertMany()
2. Select all documents in collection
3. Find the count of all suppliers
4. Find all records that have city = 'Delhi'
5. Retrieve all documents that have color equal to 'red' or 'green'
6. Retrieve all documents where part_name is 'P1' or price is less than 200.
7. Update the record of 'Geeta' ,set city = 'Bombay' and phoneno = '11223344'
8. Delete all records where price is greater than 5000
9. Display only the name and city of the supplier
10. Sort all suppliers on city and display only the first two records.

24BCA32P	**Artificial Intelligence Lab**

Prerequisites: Basic understanding of algorithms and data structures (e.g., trees, graphs, lists). Proficiency in Python programming, including libraries like NLTK for NLP tasks.

LAB Experiments

The lab experiments may be implemented in Python. Libraries like NLTK, Tensorflow and Keras may be used for Machine learning experiments.

1. Write a Python program which accepts a number and prints its prime factors.
2. Write a Python that checks whether a given password is valid.
   An acceptable password:
   a) Should have minimum 8 characters
   b) Must have at least one digit and one special character
   c) Must contain at least one alphabet which is in uppercase
3. Write a program that creates a list and performs the following operations on the list.
   a) mean, median and mode
   b) maximum and minimum values in the list
   c) Sort the list
   d) Remove duplicate values from the list
4. Create a dictionary with student names (minimum 5) as keys and a list of (subject, marks) tuples as values.
   For e.g. { "Akash": [("Math", 90), ("English", 85)…], …}
   Display student details along with total marks and percentage of each student.
5. Implement Depth-First Search (DFS) on a small graph.
6. Solve the Water Jug Problem using Breadth First Search (BFS).
7. Apply the A* Search algorithm to find the shortest path in a 4x4 grid.
8. Implement the Minimax search algorithm for 2-player games. You may use a game tree with 3 plies.
9. Implement a basic rule-based expert system for weather classification.

10. Using Python NLTK, perform the following Natural Language Processing (NLP) tasks for text content.
    a) Tokenizing
    b) Filtering Stop Words
    c) Stemming
    d) Part of Speech tagging
    e) Chunking
    f) Named Entity Recognition (NER)
11. Perform Image classification for a given dataset using CNN. You may use Tensorflow/Keras. (Not for Lab examinations)

Course outcomes:

CO1: Apply Uninformed Search Algorithms and Implement Heuristic Search techniques
CO2: Analyze and Solve Constraint Satisfaction Problems
CO3: Develop Rule-Based Systems
CO4: Implement and Evaluate Optimization Techniques
CO5: Apply and illustrate the NLP concepts

******

SEMESTER –IV

24BCA41	Computer Networks	45 Hours

Course Objectives:

CO1: Understand the fundamental concepts of Computer Networks and their applications.
CO2: Develop problem-solving skills related to network design, implementation, and troubleshooting.
CO3: Implement network protocols and configure network devices.

Prerequisites:

1. Programming Skills: Ability to write basic programs and scripts in languages such as Python or C.
2. Operating Systems: Understanding of OS concepts related to networking, such as process management and memory allocation

Course Content:

UNIT I: Introduction to Computer Networks [11 Hours]

Overview of Computer Networks: Definition and Objectives, Applications and Examples Network Components and Architecture

Network Models: OSI Model: Layers and Functions, TCP/IP Model: Layers and Functions Comparison between OSI and TCP/IP Models

Network Topologies: Physical vs. Logical Topologies, Common Topologies: Star, Ring, Bus, Mesh, Hybrid, Advantages and Disadvantages of Each Topology

Data Transmission: Analog vs. Digital Signals, Transmission Modes: Simplex, Half-Duplex, Full-Duplex, Bandwidth and Latency

Networking Devices: Routers, Switches, Hubs, Bridges, Gateways, Functions and Configurations of Each Device.

UNIT II: Data Link Layer and Networking Protocols [12 Hours]

Data Link Layer Fundamentals: Functions of the Data Link Layer, Framing, Error Detection, and Error Correction, Flow Control Mechanisms.

Ethernet: Ethernet Standards and Frame Structure, MAC Addressing and ARP, Ethernet Switching: Basic Concepts and Methods

Network Protocols: Introduction to TCP/IP Protocol Suite, IP Addressing: IPv4 and IPv6 Subnetting and CIDR Notation

Address Resolution Protocol (ARP): ARP Operation and Table, ARP Spoofing and Security Considerations

Virtual LANs (VLANs): Concept of VLANs, VLAN Tagging and Configuration, Benefits and Use Cases

UNIT III: Network Layer and Transport Layer **[11 Hours]**

Network Layer: IP Routing: Static vs. Dynamic Routing, Routing Protocols: RIP, OSPF, BGP, Network Address Translation (NAT)

Transport Layer: TCP vs. UDP: Characteristics and Use Cases, TCP Handshake and Connection Management, Flow Control and Congestion Control in TCP

Congestion Control Algorithms: Techniques: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery, TCP Variants: TCP Reno, TCP Vegas.

Network Security Fundamentals: Threats and Vulnerabilities, Basic Security Mechanisms: Firewalls, VPNs, Encryption

UNIT IV: Application Layer and Emerging Technologies [11 Hours]

Application Layer Protocols: HTTP/HTTPS: Structure and Operation, FTP, SMTP, POP3, IMAP: Protocols and Uses, DNS: Domain Name System and Resolution

Network Applications: Web Browsing, Email Communication, File Transfer, Voice over IP (VoIP) and Streaming.

Emerging Technologies: Software-Defined Networking (SDN), Network Function Virtualization (NFV), Internet of Things (IoT) and Its Impact on Networking

Text Books:

1. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2011.
2. James F. Kurose and Keith W. Ross, "Computer Networking: A Top-Down Approach", 8th Edition, Pearson, 2021.

Reference Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", 5th Edition, McGraw-Hill Education, 2012.
2. Larry L. Peterson and Bruce S. Davie, "Computer Networks: A Systems Approach", 6th Edition, Morgan Kaufmann, 2019.
3. Bhavneet Sidhu, An Integrated Approach to Computer Networks, Khanna Publishing House, 2023.
4. Mastering PC Hardware & Networking, Khanna Publishing House, 2024.

Web Resources:

1. Cisco Networking Academy - Online Courses and Resources
2. NetworkLessons.com - Tutorials on Various Networking Topics

24BCA42	Design and Analysis of Algorithms	45 Hours

Course Objectives

CO1: This course envisions to impart to students the understanding of basic algorithm designing paradigms.

CO2: This course introduces the basic knowledge on how to analyse an algorithm.

CO3: This course expects to enable a student to synthesize efficient algorithms in common design situations and real-life problems.

Prerequisite: Knowledge of Data Structures Course

Content:

UNIT I: [11 Hours]

What is an algorithm? Design and performance analysis of algorithms, time complexity, space complexity.

Asymptotic notations (O, Ω, Θ) to measure growth of a function and application to measure complexity of algorithms.

Analysis of sequential search, bubble sort, selection sort, insertion sort, matrix multiplication. Recursion: Basic concept. Analysis of recursive algorithms, Master's theorem.

UNIT II:

The Divide & Conquer Design Technique: [12 Hours]

The general concept. Binary search, finding the maximum and minimum, merge sort, quick sort. Best and worst case analysis for the mentioned algorithms. Strassen's matrix multiplication.

Lower bound for comparison-based sorting.

The Greedy Design Technique:

The general concept. Applications to general Knapsack problem, finding minimum weight spanning trees: Prim's and Kruskal's algorithms, Dijkstra's algorithm for finding single source shortest paths problem.

UNIT III: [11 Hours]

The Dynamic Programming Design Technique:

The general concept. Computation of Fibonacci series and Binomial coefficients, all pair shortest paths problem (Floyd-Warshall's algorithm), 0/1 Knapsack problem.

Algorithms on Graphs:

Breadth First Search, Depth First Search, finding connected components, depth first search of a directed graph, topological sorting.

UNIT IV: [11 Hours]

Limitations of Algorithmic Power:

Backtracking Method: n-Queen problem; sum of subsets problem/ Hamiltonian circuit problem/vertex cover problem.

Computational Intractability: Overview of non-deterministic algorithms, P, NP, NP-Complete and NP-hard problems.

Text Books

1. Levitin Anany, Introduction to Design and Analysis of Algorithms, 3rd Edition, Pearson, 2012
2. Gajendra Sharma, Design and Analysis of Algorithms, Khanna Publishing House (AICTE Recommended Textbook)
3. Cormen Thomas H., Leiserson Charles E., Rivest Ronald L. and Stein Clifford,

Introduction to Algorithms, PHI publication, 3rd Edition, 2009.
4. Horowitz Ellis, Sahni Sartaj and Rajasekaran Sanguthevar, Fundamentals of Computer Algorithms, University Press (I) Pvt. Ltd., 2012.

Reference Books

1. Aho Alfred V., Hopcroft John E. & Ullman Jeffrey D., The Design & Analysis of Computer Algorithms, Addison Wesley Publications, Boston, 1983.
2. Kleinberg Jon & Tardos Eva, Algorithm Design, Pearson Education, 2006.

Web Resources

1. https://nptel.ac.in/courses/106101060
2. https://www.cs.umd.edu/~mount/451/Lects/451lects.pdf

24BCA43	Software Engineering	60 Hours

Course Objectives

CO1: To Acquire a comprehensive understanding of the software development lifecycle and its application in contemporary software engineering practices.

CO2: To Develop proficiency in project management methodologies and strategic decision-making for successful software project execution.

CO3: To Master the art of software design, development, and testing to produce robust and efficient software solutions.

Prerequisites: Basic understand of Software, Applications, Programming fundamentals.

Course Content:

UNIT I: [15 Hours]

The evolving role of software, changing nature of software, layered technology, a process framework, Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

Agile software development: Agility Principles, Agile methods, Plan-driven and agile development, Extreme programming, Scrum, A Tool Set for the Agile Process.

UNIT II: [15 Hours]

Software Requirements Engineering: Functional and non-functional requirements, the software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Project planning- Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques.

UNIT III: [15 Hours]

Design: Design process and design quality, design concepts, the design model, software architecture, data design, architectural design, Basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT IV: [15 Hours]

Quality Management: Quality concepts, software quality assurance, software reviews, formal

technical reviews, statistical software quality assurance, software reliability.

Release Management: Release planning, development and build plans, release strategies, risk management, and post-deployment monitoring.

Product sustenance: Maintenance, updates, End of life, migration strategies.

Text Books

1. Software Engineering, N.S. Gill, Khanna Publishing House, 2023 (AICTE Recommended Textbook)
2. Software Engineering, Ian Somerville, 9th edition, Pearson education.
3. Software Engineering A practitioner's Approach, 8th edition, Roger S Pressman, Bruce R. Maxim. McGraw Hill Education, 2015.

Reference Books

1. Stephen Schach, Software Engineering 7th ed, McGraw-Hill, 2007
2. Software Engineering: Principles and Practice Hans van Vliet

24BCA41P	Computer Networks Lab

Lab Programs:

1. Configure Basic Network Settings:
   a. IP Address Configuration
   b. Subnet Mask and Gateway Settings
2. Implement Network Protocols:
   a. Write a simple Python script to perform DNS resolution.
   b. Implement a basic HTTP client-server application.
3. Network Simulation:
   a. Use network simulation tools (e.g., Cisco Packet Tracer) to design and simulate network topologies.
   b. Configure routers and switches in a simulated environment.
4. Performance Measurement:
   a. Measure network performance using tools like `ping`, `traceroute`, and `iperf`.
   b. Analyze network traffic using Wireshark.
5. Implement VLANs:
   a. Configure VLANs on a switch and verify using simulation tools.
6. Set Up a Simple Web Server:
   a. Deploy a basic web server and configure HTTP/HTTPS access.
7. Network Security Lab:
   a. Implement basic firewall rules and VPN configurations.
   b. Perform vulnerability scanning and analyze results.
8. Network Troubleshooting:
   a. Diagnose and resolve common network issues.
   b. Use troubleshooting commands and techniques to fix connectivity problems.

24BCA42P	Design and Analysis of Algorithms Lab

Solve the following problems, and find the complexity of the best and worst case:

1. Write a program to implement linear search algorithm
2. Write a program to implement binary search algorithm.
3. Write a Program to sort a given set of numbers using Selection sort algorithm.
4. Write a program to sort a given set of numbers using Bubble sort algorithm.
5. Write a program to sort a given set of numbers using Insertion sort algorithm.
6. Write a program to sort a given set of numbers using Merge sort algorithm.
7. Write a program to implement matrix multiplication.
7. Write a program to find the factorial of a number using recursive function.
8. Write a program to find the factorial of a number using recursive function.
9. Write a program to find the minimum spanning tree of a given graph using Prim's algorithm.
10. Write a program to find the minimum spanning tree of a given graph using Kruskal's algorithm.
11. Write a Program to find the binomial co-efficient C(n, k), [where n and k are integers and n > k] using brute force based algorithm and also dynamic programming based algorithm.
12. Write a program to implement BFS traversal algorithm.
13. Write a program to implement DFS traversal algorithm.

24BCASEC2	Computer Assembly and Repair Lab

1. Demonstration of Hardware peripherals: CPU, RAM, SMPS, Motherboard, NIC card, Processor, Processor cooling fan, PCI card, HDD.
2. Demonstration of various ports: CPU, VGA port, PS/2 (keyboard, mouse) ,USB, LAN, Speaker, Audio.
3. Identify the Computer Name and Hardware Specification (RAM capacity, Processor type, HDD, 32 bit/ 64 bit)
4. Identify and Troubleshoot the problems of RAM (beep sound with blue screen), SMPS and motherboard (CPU is not switched ON)
5. Configure BIOS settings- disable and enable USB and LAN.

6. Identify, how to recover the hidden files from corrupted pendrive using command.
7. Recover the contents from crashed Hard Disk using Disk Drill software.
8. Install Operating System – Windows family (Windows 7/ Windows 10) and also make partitions.
9. Install Operating System - Unix family (Linux/UBUNTU)
10. Install Application software – python 3.8, MS- Office 2010/2013, MySQL, TOAD, Openoffice, etc.,
11. Install any one of the antivirus software (Avast, Kaspersky, etc.,) and observe the variations before and after installation.
12. Add new Hardware device (keyboard, mouse, Speaker, Microphone)
13. Connect the LCD Projector with Laptop / CPU.
14. Adding additional RAM to the system.(expanding RAM size).
15. Graphic Card insertion.
16. Assemble and Disassemble Desktop System.

References:

1. Dan Gookin ,Troubleshooting & Maintaining Your PC ALL-IN-ONE, 3rd Edition,2017, John Wiley & Sons.
2. Mike Meyers, Scott Jernigan, Dan Lachance, ”CompTIA Fundamentals + Exam Guide (All-in-One), 2nd Edition, 2019, Mc Graw Hill Education.

Web References:

1. https://www.youtube.com/watch? v=ItxwyMR0SnY&list=PLeH4ngtDM7eE-1_mdWuXWyZrI_FMHnyJ0&index=5
2. https://www.cleverfiles.com/howto/crashed-hard-drive-recovery.html

SEMESTER –III

(Electives)

24BCAE1	Feature Engineering (AI & ML)	30 Hours

Prerequisite: familiarity with any programming language.

Course Content:

UNIT I: Introduction to Feature Engineering [15 Hours]

Introduction to Data and Features: Importance of Features in Machine Learning. Data types and features: Numerical, Categorical, Ordinal, Discrete, Continuous, Interval and Ratio.Basic Feature Preprocessing: Handling Missing Data, Data Cleaning, Feature Scaling, Normalization, and Transformation.

UNIT II: Feature Engineering Techniques [15 Hours]

Techniques for Numerical Data: Binning and Discretization, Polynomial and Interaction Features. Categorical Data Techniques: One Hot Encoding, Label Encoding. Feature extraction vs. feature selection, Steps in feature selection. Feature Selection Methods: Filter, Wrapper, and Hybrid. Feature Reduction: Introduction and application of Principal Components Analysis.

Text Books

1. M.C. Trivedi, Data Science and Data Analytics Using Python Programming, Khanna Publishing House, 2024.
2. Zheng, Alice, & Casari, Amanda. (2018). Feature engineering for machine learning: Principles and techniques for data scientists. O'Reilly Media, Inc.
3. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). Fundamentals of Data Science: Theory and Practice. Elsevier. ISBN-13: 9780323917780.

Reference Books:

1. Duda, R. O., Hart, P. E., Stork, D (2007). Pattern classification (2Ed), John Wiley & Sons, ISBN-13: 978-8126511167.
2. N. Bhaskar, Vasundhara, Machine Learning, Khanna Publishing House, 2024.
3. M.C. Trivedi, Deep Learning and Neural Network_MC Trivedi, Khanna Publishing House, 2024.
4. Ng, Andrew. (2018). Machine learning yearning (Draft, MIT Licensed). GitHub. ISBN- 10: 199957950X, ISBN-13: 978-1999579500.
5. Han, Jiawei, Kamber, Micheline, & Pei, Jian. (2011). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann Publishers. ISBN 978-0123814791.
6. Tan, Pang-Ning, Steinbach, Michael, Karpatne, Anuj, & Kumar, Vipin. (2021). Introduction to data mining (2nd ed.). Pearson. ISBN 978-9354491047.
7. Provost, Foster, & Fawcett, Tom. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.
8. Galli, Soledad. (2020). Python feature engineering cookbook: Over 70 recipes for creating, engineering, and transforming features to build machine learning models. Packt Publishing, Limited.

9. Nielsen, Aileen. (2019). Practical time series analysis: Prediction with statistics and machine learning. O'Reilly Media.
10. Rajiv Chopra, Deep Learning, Khanna Publishing House, 2024.
11. Jeeva Jose, Machine Learning, Khanna Publishing House, 2024.
12. Chollet, François. (2017). Deep learning with Python. Manning Publications. ISBN 9781617294433.

Course Outcomes

CO1:Understand the importance of features in machine learning and differentiate between various types of data and features (structured vs. unstructured, categorical, numerical, text, and date-time).
CO2: Apply basic feature preprocessing techniques such as handling missing data, data cleaning, and feature scaling and normalization.
CO3:Implement feature engineering techniques for numerical data, including binning, discretization, polynomial and interaction features, and log transformation.
CO4:Utilize categorical data techniques, such as one-hot encoding and label encoding, and understand feature selection methods, including filter and wrapper methods.
CO5: Perform feature transformation using techniques like Principal Component Analysis (PCA) and understand its application in machine learning.

******

24BCAE1	**Basics of Data Analytics using Spreadsheet (Data Science)**	**30 Hours**

Course Objectives

CO1: Understand the basics of data analytics and its applications.
CO2: Develop proficiency in using spreadsheet software for data manipulation and analysis.
CO3: Build and use spreadsheet models for decision making & Communicate data insights effectively

Prerequisite: Knowledge on basics of mathematical & Statistical concepts such as arithmetic, percentages, averages, and basic algebra.

Course Content:

UNIT I: Introduction to Data Analytics **[15 Hours]**

Understanding data and its types (structured, unstructured, semi-structured)-What is Data Analytics- Types of data Analytics-Importance of Data Analytics- Applications of Data Analytics.

UNIT II: Data, Ethics, and Industry: Case Studies [15 Hours]

Data Collection Methods - Different Data Sources & format - Data Cleaning and Transformation - Handling Missing Data and Outliers. - Ethical considerations in data analytics. - Real-world Applications of Data Analytics- Industry-specific applications (finance, marketing, operations) - Case Study

Note: Case study is for discussion not to be considered for evaluation.

Text Books

1. “Beginner's Guide for Data Analysis using R Programming” by Jeeva Jose, Khanna Publishing House, 2024.
2. “Data Analytics” by V.K. Jain, Khanna Book Publishing Company, 2024.
3. “Excel Data Analysis For Dummies" by Stephen L. Nelson and E. C. Nelson, John Wiley & Sons; 3rd edition, 2016
4. "Data Analysis Using Microsoft Excel" by Michael R. Middleton, Thomson, Brooks/Cole, 3rd edition , 2004

Reference Books

1. "Excel 2019 Bible" by Michael Alexander, Richard Kusleika, and John Walkenbach, John Wiley & Sons, 25 Sept 2018
2. "Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics" by Cliff T Ragsdale, Cegage learning asia pet. 2015
3. “Mastering Excel” by WebTech Solutions, Khanna Publishing House, 2024.

****

24BCAE1	Web Programming-I (Full Stack Development)	30 Hours

Course Objectives

1. To understand the concepts and architecture of the World Wide Web, Markup languages along with Cascading Style Sheets.
2. To understand the concepts of event handling and data validation mechanisms.
3. To understand the concepts of embedded dynamic scripting on client and server-side Internet Programming and basic full stack web development.
4. To develop modern interactive web applications

Prerequisite:

1. Proficiency in at least one programming language, such as C, C++. Understanding of programming concepts such as loops, conditionals, functions, and data structures like arrays, lists.
2. Familiarity with Object-Oriented Programming (OOP) principles, including classes, objects, inheritance, and polymorphism.

UNIT I: [15 hours]

Introduction to HTML: history of HTML, Objective, basic Structures of HTML, Header Tags, body tags, Paragraph Tags.
Tags for FORM Creation, TABLE, FORM, TEXTAREA, SELECT, IMG, IFRAME FIELDSET, ANCHOR.
Lists in HTML, Introduction to DIV tag, NAVBAR Design.
Introduction to CSS, types, Selectors, and Responsiveness of a web page.
Introduction to Bootstrap, downloads/linking, using classes of Bootstrap, understanding the Grid System in Bootstrap.
Introduction to www, Protocols and Programs, Applications and development tools, web browsers, DNS, Web hosting Provider, Setting up of Windows/Linux/Unix web servers, Web hosting in cloud, Types of Web Hosting.

UNIT II: [15 hours]

Introduction to JavaScript: Functions and Events. Document Object model traversing using JavaScript. Output System in JavaScript i.e. Alert, throughput, Input box, Console. Variables and Arrays in JavaScript. Date and String handling in JavaScript.
Manipulating CSS through JavaScript: Form Validation like Required validator, length validator, Pattern validator. Advanced JavaScript, Combining HTML, CSS and JavaScript events and buttons, controlling your browser.
Introduction to AJAX: Purpose, advantages and disadvantages, AJAX based Web applications and alternatives of AJAX.
Introduction to XML: uses, Key concepts, DTD schemas, XSL, XSLT, and XSL Elements and transforming with XSLT. Introduction to XHTML.
JSON: Introduction to JSON, Keys and Values, Types of Values, Arrays, Objects

Text Books

1. Laura Lemay, Mastering HTML, CSS & Java Script Web Publishing, BPB Publications, 2016
2. Thomas A. Powell, The Complete Reference HTML & CSS, Fifth Edition, 2017

Reference Books

1. Silvio Moreto, Bootstrap 4 By Example, ebook, 2016.

Web Resources

1. www.javatpoint.com
2. www.w3schools.com
3. https://www.geeksforgeeks.org/web-technology/

****

SEMESTER –IV

(Electives)

24BCAE2	Introduction to Machine Learning (AI&ML)	30 Hours

Prerequisites: Basic knowledge of statistics and probability. Familiarity with fundamental programming concepts and proficiency in Python.

Course Content:

UNIT I: Introduction to Machine Learning [15 Hours]

Introduction: Definition, History and Application of Machine Learning, *Types of Machine Learning*: Supervised, Unsupervised, Semi-Supervised, and Reinforcement Learning. Labeled and Unlabelled Dataset. *Supervised Learning Tasks*: Regression vs. Classification, *Learning Framework*: Training, Validation and Testing of ML models. *Performance Evaluation Parameters*: Confusion matrix, Accuracy, Precision, Recall, F1 Score, and AUC.

UNIT II: Supervised Learning and Unsupervised Learning [15 Hours]

Regression: Linear and non-linear Regression, Logistic Regression. *Classification*: Naïve Bayes, K-Nearest Neighbors, *Clustering*: K-Means, Hierarchical Clustering, DBSCAN, Clustering Validation Measures. *ML Applications:* Ethical Considerations in Machine Learning, Case study and Real-world Applications.

Text Books:

1. Rajiv Chopra (2024), Machine Learning and Machine Intelligence, Khanna Publishing House.
2. Jeeva Jose (2023), Introduction to Machine Learning, Khanna Publishing House.
3. Mitchell T. (1997). Machine Learning, First Edition, McGraw-Hill.
4. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). Fundamentals of Data Science: Theory and Practice. Elsevier. ISBN9780323917780

Reference Books:

1. Flach, P. A. (2012). Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press. ISBN: 9781107422223, 2012.
2. Duda, R. O., Hart, P. E., Stork, D (2007). Pattern classification (2Ed), John Wiley & Sons, ISBN-13: 978-8126511167.
3. Haykin S. (2009). Neural Networks and Learning Machines, Third Edition, PHI Learning.
4. Chollet, F. (2018). Deep Learning with Python. Manning Publications.
5. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
7. Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (1st ed.). O'Reilly Media.

Course Outcomes

CO1: Define and explain machine learning concepts, types, and basic metrics.
CO2:Implement and apply supervised learning techniques (e.g., KNN, Linear Regression,

Logistic Regression).

CO3: Apply unsupervised learning methods (e.g., K-Means, Hierarchical Clustering, Association Rules).

CO4: Develop and evaluate simple machine learning models (e.g., Perceptron, single-layer neural networks).

CO5: Analyze and apply appropriate machine learning algorithms depending on the problems with some real-world data.

24BCAE2	Data Visualization (Data Science)	30 Hours

Course Objectives

CO1:Understand the fundamentals of data visualization and its importance.
CO2:Learn about visual perception and its impact on data interpretation.
CO3:Explore the ethical considerations and challenges in data visualization.
CO4:Study different types of visualizations and their appropriate uses.
CO5:Utilize Power BI to create and customize various types of visualizations.

Prerequisite:

Familiarity with using a computer, including file management and basic software navigation. Basic knowledge of data structures, such as tables and databases. Basic understanding of data analysis concepts and familiarity with data types.

Course Content:

UNIT I: Introduction to Data Visualization [15 Hours]

Definition and importance of data visualization-Role of data visualization in decision making-Types of data (numerical, categorical, temporal, geographical)-Data visualization process (data collection, exploration, analysis, visualization, interpretation)-Challenges and limitations of data visualization

UNIT II: Visualization tools & Data Storytelling [15 Hours]

Overview of Visualization Tools (e.g., Excel, Tableau, Power BI, Python)- Comparing and contrasting features and Use Cases among these tools.

Principles of Data Storytelling: Narrative and Context-Best Practices for Dashboard Layout and Interactivity

Principles of Good Visualization Design - Understanding and Using Color in Visualizations – Importance of Data Modelling in Visualization.

Text Books

1. "Storytelling with Data: A Data Visualization Guide for Business Professionals" Cole Nussbaumer Knaflic, Wiley; 1st edition, 2015.
2. “The Visual Display of Quantitative Information” by Edward Tufte, Graphics Press USA; 2nd edition, 2001.

Reference Books

1. "Data Visualization: A Practical Introduction" Kieran Healy, Princeton University Press, 2018.
2. "Analyzing Data with Power BI and Power Pivot for Excel", Alberto Ferrari and Marco Russo, Microsoft Press; 1st edition, 2017.
3. "Microsoft Power BI Complete Reference", Devin Knight, Brian Knight, Mitchell Pearson, and Manuel Quintana, Packt Publishing; 1st edition, 2018.

******

24BCAE2	Web Programming-II (Full Stack Development)	30 Hours

Course Objectives

1. To develop dynamic web applications that can interact with users and respond dynamically using PHP
2. To implement server-side scripting to perform operations like validations, calculations, and content generation on the server
3. To use PHP with databases to perform CRUD operations, and display data on web pages
4. To implement session and cookie handling
5. To develop secure web applications such as input validation, sanitization, and protection against SQL injection.
6. To integrate PHP with HTML, CSS, and JavaScript
7. To understand file handling operations Read, write, upload, and manage files using PHP scripts.

Prerequisite:

1. Understand HTML tags, forms, inputs, and structure of a webpage.
2. Familiarity with CSS
3. Understanding of Basic Programming Concepts
4. Basics of Client-Server Architecture

UNIT I: [15 hours]

Introduction to PHP: Server-Side Scripting – Advantages of PHP – Installation & Configuration. Evaluation of PHP, Basic Syntax, Defining variable and constant, PHP Data types, Operators and Expressions

Decisions and Loop: Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with HTML.

Function: Define a function, Call by value and Call by reference, Recursive function.

String: Creating and accessing, String Searching & Replacing String, Formatting String, String Related Library function

Array: Anatomy of an Array, Creating index based and Associative array, Accessing array, Element Looping with Index based array, Looping, Some useful Library function

Exception Handling: Understanding Exception and error, try, catch, throw. error tracking and debugging

UNIT II: [15 hours]

Handling HTML Form with PHP: Capturing Form, Data Dealing with Multi-value filed, and Generating File uploaded form, redirecting a form after submission.

Working with file and Directories: Understanding file & directory, Opening and closing, a file, Copying, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.

Session and Cookie: Introduction to Session Control, Session Functionality, Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Modify Cookies, Deleting Cookies, Registering

Session variables, Destroying the variables and Session.

Database Connectivity with MySQL: Connection with MySQL Database, performing basic database operation (DML:Insert, Delete, Update, Select), Setting query parameter, Executing query Join (Cross joins, Inner joins, Outer Joins, Self joins.)

Text Books

1. PHP and MySQL Web Development - Welling – (Pearson Education)-Fourth Edition
2. Learning PHP, MySQL, books by ‘ O’ riley Press
3. Core PHP Programming. Leon Atkinson, Prentice Hall.

Reference Books

1. Programming PHP. RasmusLerdorf, Kevin Tatroe. O'Reilly.
2. Learning PHP 5. David Sklar, O'Reilly.

Web Resources

1. www.w3schools.com
2. https://www.geeksforgeeks.org/web-technology/

****