Scheme of Instructions & Syllabi
of
Bachelor of Computer Applications
(Artificial Intelligence)
1st Year
(2025-28)

Total Credit of the Program

Semester	I	II	III	IV	V	VI	Total
Credits	24	27	26	30	24	24	**155**

Dr. Akash Sanghi	Prof. YDS Arya	Prof. Manish Gupta
HOD, Computer Applications	Vice Chancellor	Dean, Computer Applications

Faculty of Computer Applications
INVERTIS UNIVERSITY
Bareilly-243123 U.P.

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER I, YEAR I

S.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	BCAICC101	English-I	2+1+0	25	50	75	3
2	BCAICC102	Theory of Mathematics	2+1+0	25	50	75	3
3	BCAICC103	Programming in C	3+1+0	30	70	100	4
4	BCAICC104	Client Side Scripting	3+1+0	30	70	100	4
PRACTICAL / PROJECTS							
5	IIOT1AC	Data Analytics	0+0+4	30	70	100	4
6	BCAICC106	Office Automation Lab	0+0+2	10	15	25	1
7	BCAICC107	Programming in C Lab	0+0+4	15	35	50	2
8	BCAICC108	Client Side Scripting Lab	0+0+4	15	35	50	2
9	BCAICC109	Effective Communication Skills	0+0+2	10	15	25	1
TOTAL			**10+4+16**	**190**	**410**	**600**	**24**

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER II, YEAR I

S.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	BCAICC201	Linux Shell Scripting	3+1+0	30	70	100	4
2	BCAICC202	Python Programming	3+1+0	30	70	100	4
3	BCAICC203	Power BI	2+1+0	25	50	75	3
4	BCAICC204	Operating Systems	2+1+0	25	50	75	3
5	BCAICC205	Environmental Studies	2+1+0	25	50	75	3
PRACTICAL / PROJECTS							
6	IIOT2AC	Advanced Data Analytics	3+1+0	30	70	100	4
7	BCAICC251	Linux Shell Scripting Lab	0+0+4	15	35	50	2
8	BCAICC252	Python Programming Lab	0+0+4	15	35	50	2
9	BCAICC253	Power BI Lab	0+0+4	15	35	50	2
TOTAL			**15+6+12**	**210**	**465**	**675**	**27**

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER III, YEAR II

S.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	BCAICC301	Database Management Systems	3+1+0	30	70	100	4
2	BCAICC302	Data Structures	3+1+0	30	70	100	4
3	BCAICC303	Data Communication and Computer Networks	3+1+0	30	70	100	4
4	BCAI304	Statistics and Probability	2+1+0	25	50	75	3
5		Elective-I	2+1+0	25	50	75	3
PRACTICAL / PROJECTS							
6	IIOT3AC	Machine Learning	0+0+4	30	70	100	4
7	BCAICC351	Database Management Systems Lab	0+0+4	15	35	50	2
8	BCAICC352	Data Structures Lab	0+0+4	15	35	50	2
TOTAL			**13+5+12**	**200**	**450**	**650**	**26**

Elective I		
S.No	**Course Code**	**Program Elective-I**
1	BCAI305	Digital Electronics
2	BCAI306	Data Visualization

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER IV, YEAR II

S.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	BCAI401	Introduction to Artificial Intelligence	3+1+0	30	70	100	4
2	BCAI402	Sensor Technology	2+1+0	25	50	75	3
3		Elective-II	2+1+0	25	50	75	3
4	BCAICC406	Responsive Web Design-Front End Development	3+1+0	30	70	100	4
5	BCAICC407	Java Programming	3+1+0	30	70	100	4
6	BCAICC408	Logical Reasoning and Thinking	3+1+0	15	35	50	2
PRACTICAL / PROJECTS							
6	IIOT4AC	Deep Learning	0+0+4	30	70	100	4
7	BCAI451	Introduction to Artificial Intelligence Lab	0+0+4	15	35	50	2
8	BCAICC456	Responsive Web Design-Front End Development Lab	0+0+4	15	35	50	2
10	BCAICC457	Java Programming Lab	0+0+4	15	35	50	2
TOTAL			**16+6+16**	**230**	**520**	**750**	**30**

Elective II		
S.No	**Course Code**	**Program Elective-II**
1	BCAI403	Business Process Management
2	BCAI404	Introduction to Data Science
3	BCAI405	Pattern Recognition

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER V, YEAR III

S.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
1	BCAI501	MERN Full Stack Development- Backend	3+1+0	30	70	100	4
2	BCAI502	Block Chain Technology	3+1+0	30	70	100	4
3		Elective –III	2+1+0	25	50	75	3
4		Elective – IV	2+1+0	25	50	75	3
5		Generic Elective-I	1+1+0	15	35	50	2
PRACTICAL / PROJECTS							
6	IIOT5AC	Advanced Artificial Intelligence	0+0+4	30	70	100	4
7	BCAI551	MERN Full Stack Development- Backend Lab	0+0+4	15	35	50	2
8		Elective – III Lab	0+0+4	15	35	50	2
TOTAL			**11+5+12**	**185**	**415**	**600**	**24**

Elective III		
S.No	**Course Code**	**Program Elective-III**
1	BCAI513	Server-Side Scripting Languages
2	BCAI514	Android Application Development

Elective III Lab		
S.No	**Course Code**	**Program Elective-III Lab**
1	BCAI553	Server-Side Scripting Languages Lab
2	BCAI554	Android Application Development Lab

Elective IV		
S.No	**Course Code**	**Program Elective-IV**
1	BCAI515	Introduction to Web Services
2	BCAI516	Cyber Law and Information Security
3	BCAI517	Embedded Systems

Generic Elective-I		
S.No	**Course Code**	**Generic Elective-I**
1	BCAI518	Soft Computing
2	BCAI519	Cloud Computing
3	BCAI520	Client Server Computing

STUDY AND EVALUATION SCHEME

Bachelor of Computer Applications (Artificial Intelligence)

(Session 2025-28)

SEMESTER VI, YEAR III

.No	Course Code	Course Title	L+T+P	CA	EE	Total	Credit
PRACTICAL / PROJECTS							
1	BCAICC601	Internship	0+0+0	180	420	600	24
TOTAL			**0+0+0**	**180**	**420**	**600**	**24**

Syllabus

Semester: I

BCAICC101: English-I						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of English Grammar and Vocabulary, Fundamental Interpersonal and Presentation Skills

Course Objectives:

1. To enhance students' proficiency in English language skills—listening, speaking, reading, and writing—for professional contexts.
2. To develop effective verbal and non-verbal communication skills required in workplace settings, including interviews, meetings, and presentations.
3. To familiarize students with business and technical communication formats such as reports, emails, proposals, and notices.
4. To cultivate interpersonal skills, confidence, and etiquette necessary for teamwork, leadership, and cross-cultural communication.

Detailed Syllabus

Unit I: Communication Process

What is communication? The communication model, elements of communication, Importance of effective communication skills in the business world, Components of Communication Process, practicing effective communication, good communication Vs effective communication, styles of communication, intercultural communication skills- need for attitude change and benefits

Unit II: Types of Communication & Barriers to communication

Verbal Communication, Non Verbal Communication, Written Communication, Do's and don'ts of each type, barriers to effective communication and how to overcome them, interaction of verbal and non-verbal communication, talents of a corporate communicator, silence- merits and limitations of each type

Unit III: Listening Skills & Reading Skills

What is listening, various types of listening – Active, passive, selective, listening and note taking, listening and comprehending, listening to speak, principles of good listening Techniques to develop effective listening skills, Reading Skills- skimming, scanning and inferring- common reading techniques, practicing smart reading.

Unit IV: Conversation Skills

Importance of conversation skills, features of a good conversation, Tips to improve Conversation skills, importance of questioning skills, techniques to ask right questions- role play situations to practice the same, discussing issues (social, political and cultural), formal and informal conversation

Unit V: Telephone Etiquette

Basic rules of telephone etiquette- formal vs. informal; tone, pitch and vocabulary related to formal ways of speaking over the phone, leaving voice messages; practice sessions (role plays)

Persuasive communication: What is persuasive communication, different techniques of persuasive communication, How to negotiate using persuasive communication, the act of negotiation, negotiation style and their contexts, fundamentals of negotiation, common hurdles in negotiation and how to overcome them

Text Books:

1. "Active Listening 101: How to Turn Down Your Volume to Turn Up Your Communication Skills, by Emilia Hardman, 2012
2. The Power of Communication: Skills to Build Trust, Inspire Loyalty, and Lead Effectively, by Helio Fred Garcia, 2012

Reference Books:

1. Power Listening: Mastering the Most Critical Business Skill of All, by Bernard T. Ferrari, 2012
2. Fitly Spoken: Developing Effective Communication and Social Skills, by Greg S. Baker, 2011
3. The Secrets of Successful Communication: A Simple Guide to Effective Encounters in Business (Big Brain vs. Little Brain Communication), by Kevin T. McCartney, 2011.

Course Outcomes:

CO1: Demonstrate proficiency in English language skills including listening, speaking, reading, and writing for professional and academic purposes.
CO2: Apply principles of effective verbal and non-verbal communication in interviews, group discussions, and workplace interactions.
CO3: Write clear, concise, and well-structured professional documents such as emails, reports, proposals, and notices.
CO4: Deliver effective presentations and participate confidently in meetings and public speaking situations using appropriate communication tools and techniques.
CO5: Exhibit professionalism, teamwork, and intercultural communication skills essential for career development and workplace success.

BCAICC102: Theory of Mathematics						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of High School Mathematics, Familiarity with Logical Reasoning and Problem Solving

Course Objectives:

1. To develop a solid foundation in mathematical logic, set theory, relations, functions, and proof techniques essential for theoretical computer science.
2. To enhance analytical thinking and problem-solving abilities through the study of algebraic structures, combinatorics, and number theory.
3. To provide learners with the tools to understand and apply mathematical reasoning in computer algorithms, automata theory, and programming logic.
4. To introduce concepts of matrices, linear algebra, and graph theory that are crucial for data science, machine learning, and advanced computing applications.

Detailed Syllabus

Unit I: Sets, relations and functions
Definition of Set, Type of Sets, Operations on Sets, Venn diagram, Cartesian Product, Relations, Functions, Types of function, Some elementary functions with their graphs (Exponential, logarithmic, modulus), Limit & continuity of a function (Simple Problems).

Unit II: Matrices and Determinants
Types of Matrices, Operations of addition, Scalar Multiplication and Multiplication of Matrices, Determinant of a Square Matrix, Minors and Cofactors, Transpose, adjoint and inverse of a matrix, solving system of linear equations, in two or three variables using inverse of a matrix.

Unit III: Limits, Continuity and Differentiation
Limit at a Point, Properties of Limit, Computation of Limits of Various Types of Functions, Continuity at a Point, Continuity Over an Interval, Intermediate Value Theorem, Type of Discontinuities. Derivative, Derivatives of Sum, Differences, Product & Quotients, Chain Rule, Derivatives of Composite Functions

Unit IV: Coordinate Geometry
2D Cartesian Co-ordinate system, Straight line: (Equation & Slope of a line), Circle: Equation of Circle, Equation to Tangent, Conic Sections: Focus, Eccentricity, Directrix, Axis of a conic section, Parabola & Ellipse: (Definitions, equations and shape of curve only)

Unit V: Statistics and Probability
Introduction, definition, terminologies in statistics and probability, measure of central tendency and dispersion, probability distribution – continuous and discrete, Bayes theorem, testing of hypothesis, basics of ANOVA, Correlation and Regression Analysis.

Text Book:

1. Mathematics for BCA by G. C. Sharma & Madhu Jain, Oscar Publication
2. Mathematics Vol-2 by R. D. Sharma, Dhalpat Raj & Sons

Reference Books:

1. The Elements of Co-ordinate Geometry Part-I by S. L. Loney, Book Palace, New Delhi.

Course Outcome:

After completion of the course the student will be able:

CO1: Understand and apply fundamental concepts of sets, logic, and proof techniques in mathematical and computational problem solving.
CO2: Analyze relations and functions, and use them to model and solve real-life and theoretical problems in computer science.
CO3: Demonstrate proficiency in algebraic structures such as groups, rings, and fields relevant to cryptography and coding theory.
CO4: Apply combinatorics and number theory in algorithm analysis, discrete probability, and network theory.
CO5: Utilize matrices, determinants, and linear algebra in data representation, system of equations, and applications in computer graphics and data science.

BCAICC103: Programming in C						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Boolean Algebra, Number System and basic mathematical formulas

Course Objectives:

1. To learn fundamentals of C Programming, Operators and Control Statements in C.
2. To understand usage of loop statements, arrays and strings in C Programming.
3. To learn the fundamentals of pointers, usage of pointers, memory allocation and functions in C Programming.
4. To understand the visibility and scope of variables on the basis of storage classes and defining user defined data types using structure, unions and enums in C Programming.
5. To learn how to handle a file using C Programs and fundamentals of an Operating System (Linux) Programming.

Detailed Syllabus

Unit I: Introduction to Programming:
Problem Solving Using Computers: Language Classification, Problem Analysis, Algorithm and Flowchart design. Algorithms: Steps in developing algorithms, advantages and disadvantages. Flowcharts: Symbols used in developing flowcharts, advantages and disadvantages. Coding, testing, debugging, Documentation and maintenance. Program development and modular design.

Unit II: Fundamentals of C programming and Control Structures:
History, Structure of a C program, C Conventions, Character Set, Identifiers, Keywords, Simple Data types, Modifiers, Variables, Constants, Operators (Arithmetic operator, relational operator, logical operator, ternary operator, unary operator, shorthand operator, bit-wise operator and arithmetic operator) Operator precedence.

Unit III: Input and Output operation:
Input and Output operation: Single character input and output, formatted input and output, Buffered input. Control Structures: Introduction, Conditional statement, if statement, if-else statement, nested if statement, else-if statement and switch statement. Goto statement. Looping statement, while statement, do-while statement, for statement, break and continue, nested for statement

Unit IV: Arrays and Functions:
Arrays: Introduction (One and multi-dimensional), Declaration of arrays, Initialization of arrays, processing with arrays. String manipulation, declaration of string arrays, string operations.
Functions: Introduction, advantages of subprograms, Function definition, function call, Actual and formal arguments, local and global variables, function prototypes, types of functions, recursive functions, arrays and functions.

Unit V: Storage Classes, Structures, Unions and Pointers:
Storage Classes, Structures and Unions: Introduction, types of storage classes, Introduction to structures, Advantages of structures, accessing elements of a structure, nested structures, array of structures, functions and structures, Unions, bit-fields, enumerated data types.
Pointers: Introduction, pointer variable, pointer operator, pointer arithmetic, pointers and arrays, pointers and strings, array pointers, dynamic allocation

Unit VI: Files, Preprocessor, standard library and header files:
Files: Introduction, File data type, opening and closing a file, file functions (getc, putc, getw, putw, fscanf, fprintf, fread, fwrite, fgets, fputs, feof).
Preprocessor: #define, #include, #undef, Conditional compilation directives, C standard library and header files: Header files, string functions, mathematical functions, Date and Time functions

Text Book:

1. Forouzon A Behrouz, Gilberg F Richard, A Structured Programming Approach using C - 3rd Illustrated Edition, 2009.
2. Kanetkar, Yashavant: “Let Us C”, 6th Edition. BPB Publications.
3. Balagurusamy, E: “Programming in ANSI C” 3rd Edition. Tata McGraw-Hill

Reference Books:

1. The C programming Language by Richie and Kernighan, 2004, BPB Publication
2. Gottfried, Byron S: “Programming with C”, 1996. Tata McGraw-Hill
3. Deitel, H M and Deitel P J: “C How to Program”, 2nd Edition. Prentice-Hall

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamentals of C programming, including data types, operators, expressions, and input/output operations.
CO2: Apply control structures such as decision-making, looping, and branching to solve computational problems.
CO3: Develop programs using functions, recursion, and arrays for efficient modular programming.
CO4: Utilize pointers, structures, and unions to handle complex data manipulation and memory management.
CO5: Implement file handling techniques to read from and write to files for data storage and retrieval.
.

BCAICC104: Client Side Scripting						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Basic Knowledge of Computer Fundamentals, Familiarity with HTML and Web Browsers

Course Objectives:

1. To introduce the fundamental concepts of client-side web development
2. To enable students to understand and implement dynamic behavior on web pages
3. To develop skills for validating forms and handling events on the client side
4. To familiarize students with modern scripting practices and debugging techniques

Detailed Syllabus

Unit I: Introduction to the Internet and the World Wide Web
Introduction, History of internet, Internet Design Principles, Internet Protocols - FTP, TCP/IP, SMTP, Telnet, etc., Client Server Communication, Web System architecture, Evolution of the Web, Web architectures, Web clients and servers, Static and Dynamic Web Applications, Front end and back end web development. HTML, CSS, JS, XML; HTTP, secure HTTP, etc.; URL, Web Services – SOAP, REST.

Unit II: HTML, CSS & JavaScript
Introduction to Html, Html Document structure, Html Editors, Html element/tag & attributes, designing simple page - Html tag, Head tag, Body tag; More Html tags - Anchor tag, Image tag, Table tag, List tag, Frame tag, div tag

Unit III: Html forms
Html forms - Input type, Text area, Select, Button, Images. Introduction to CSS, Syntax, Selectors, Embedding CSS to Html, Formatting fonts, Text & background color, Inline styles, External and Internal Style Sheets, Borders & boxing. Introduction to JavaScript: Data types, variables, operators, expressions, statements, functions, objects, arrays, date, math, error handling, flow control, loops

Unit IV: XML and HTML5, CSS3
Introduction to XML, Difference b/w Html & XML, XML editors, XML Elements & Attributes XML DTD, XML Schema, XML Parser, Document Object Model (DOM), XML DOM. Introduction to HTML5, CSS3, New features, Local storage, Web Sockets, Server events, Canvas, Audio & Video, Geolocation, Microdata, Drag and Drop. Browser life cycle and browser rendering stages. Service workers.

Unit V: Practical website development
Commonly used Web Servers and browsers, Setting up a server and domain name, website types and structures, web authoring tools, Web hosting, website maintenance, generating traffic to your website.

Unit VI: PHP Server side scripting

Introduction to PHP, Basic Syntax, Variables, constants and operators, Loops, Arrays and Strings, Environment & environment variables, responding to HTTP requests, Files, Cookies, Sessions, Examples.

Text Books:

1. Practical Web Design for Absolute Beginners, Adrian W. West. Apress 2016
2. Introducing Web Development, Jorg Krause. Apress 2017.
3. HTML & CSS: The Complete Reference, Thomas Powell. McGraw Hill, Fifth Edition, 2010
4. Creating a Website: The Missing Manual, 3rd Edition, Mathew Macdonald. O'Reilly
5. Web Technologies - HTML, JavaScript, PHP, Java, JSP, ASP.NET, XML and Ajax Black, Kogen Learning Systems (Dreamtech Press), 5th Edition 2009.

Reference Books:

1. HTML, XHTML & CSS Bible, Brian Pfaffenberger, Steven M.Schafer, Charles White, Bill Karow- Wiley Publishing Inc, 2010
2. HTML5 & CSS3 for the Real World, 2 Edition, Alexis Goldstein, Estelle Weyl, Louis Lazaris. Apress 2015.
3. HTML5 & CSS3 for Dummies, Andy Harris. Wiley 2014.
4. Learning PHP A Gentle Introduction to the Web's Most Popular Language, David Sklar. O'Reilly 2016.
5. Build Your Own Database Driven Web Site Using PHP & MySQL, Kevin Yank. Sitepoint , 4th Edition, 2009.

Course Outcome:

After completing the course, students will be able to:

CO1: Develop interactive and responsive web pages using HTML, CSS, and JavaScript.
CO2: Apply JavaScript to manipulate Document Object Model (DOM) elements dynamically.
CO3: Implement client-side form validation and handle user-triggered events effectively.
CO4: Use browser developer tools to debug and optimize client-side scripts.
CO5: Design user-friendly interfaces through efficient scripting practices and techniques.

Syllabus

Semester: II

BCAICC201: Linux Shell Scripting						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Basic Knowledge of Operating Systems, Familiarity with Linux Command Line Interface (CLI)

Course Objectives:

1. Understand the fundamentals of Linux operating system and shell environments.
2. Learn to write, execute, and debug shell scripts for task automation.
3. Develop skills to manage files, processes, and system operations using shell commands.
4. Apply scripting techniques to solve real-world system administration problems.

Detailed Syllabus

Unit 1: Introduction to UNIX:
History of UNIX - Unix Components/Architecture - Features of Unix – UNIX Environment and UNIX Structure - Posix and Single Unix specification - The login prompt - UNIX commands – Basic commands - echo, printf, ls, who, date,passwd, cal - Combining commands
Internal and external commands – type, man, more and other commands - theuser terminal, displaying its characteristics and setting characteristics - The root login - super user: sucommand - /etc/passwd and /etc/shadow files - Commands to add, modify and delete users.

Unit II: UNIX file system:
UNIX File basics - File types and Categories – File Organization – Directories - home directory and the HOME variable - Reaching required files- the PATH variable - Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent parent directories - File related commands – cat, mv, rm, cp, wc

Unit III: File Management:
File inodes and the inode structure. File links – hard and soft links – Head and tail commands
Cut and paste commands - The sort command - Special files /dev/null and /dev/tty - File attributes and permissions - The umask and default file permissions - ls command - Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions

Unit IV: UNIX Process Management:
The Structure of Processes: Process States and Transitions - Layout of system memory - Context of a process. Process Control: Process Creation – Signals – Process Termination – Invoking other programs – PID & PPID – Shell on a Shell.

Unit V: Vi Editor:
Introduction to Text Processing, Command & edit Mode, Invoking vi, deleting & inserting Line, Deleting & Replacing Character, Searching for Strings, Yanking, Running Shell Command Macros, Set Window, Set Auto Indent, Set No. Communicating with Other Users: who, mail, wall, send, mesg.

Unit VI: Shell programming:
Introduction – Need for Scripts – Creating and Calling the Script – The Shebang – Different ways of running a script - Using variables in Script – Reading Input – Integer Variables – Arithmetic Expressions – Read-only variables – Exporting variables – Arrays - Control Statements: If, Then, Else, While and Until, Classic For, Break and Continue, Case – Handling Script Parameters: Shift, Getopts – Shell Functions – Handling Conditional expression patterns and Regular expressions in scripts.

Text Books:

1. Sumitabha Das., UNIX Concepts and Applications. 4th Edition. Tata McGraw Hill, July 2017.
2. Behrouz A. Forouzan, Richard F. Gilberg : UNIX and Shell Programming- Cengage Learning – India Edition. 2009

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
2. Richard Blum, Christine Bresnahan: Linux Command Line and Shell Scripting Bible, 2nd Edition , Wiley,2014.

Course Outcomes:

After completing the course, students will be able to:

CO1: Demonstrate proficiency in using Linux commands and shell environments.
CO2: Write efficient shell scripts to automate routine system tasks.
CO3: Manage files, directories, and processes through script-based solutions.
CO4: Apply conditional statements, loops, and functions in shell programming.
CO5: Troubleshoot and debug shell scripts to ensure smooth execution.

BCAICC202: Python Programming						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Basic computer skills, a problem-solving mindset, and familiarity with programming concepts

Course Objectives:

1. To know the basics of Programming
2. To construct Python programs with control structures.
3. To structure a Python Program as a set of functions.
4. To use Python data structures-lists, tuples, dictionaries.
5. To do input/output with files in Python.

Detailed Syllabus

Unit I: Introduction to Python

Introduction: Introduction to Python, setting up the environment, Installing Python, running python program, Python's execution model, Guidelines on how to write good, The Python culture, A note on the IDEs. **Built-in Data Types:** Numbers, Immutable sequences, Mutable sequences, set types, Mapping types – dictionaries, the collections module, Final considerations **Iterating and Making Decisions:** Conditional programming, Looping, Putting this all together.

Unit II: Advanced Concepts

Program Control Flow Conditional Statements The if Statement, The if-else Statement, The if-elif Statement, The if-elif Statement, Nested if Statements, Python Indentation, Looping and Iteration, The For Loop, The While Loop, Loop else Statement, Nested Loops, Break and Continue, The Range Function, Introduction to range(), Types of range() function, Use of range() function.

Introduction To Functions Built-In Functions, Introduction to Functions, Using a Functions, Python Function Types, Structure of Python Functions, E.g. - map, zip, reduce, filter, any, chr, ord, sorted, globals, locals, all, etc.

User Defined Functions Structure of a Python Program w.r.t. UDF, Types of Functions, Invoking UDF, Flow of Execution, Arguments and Parameters, Default Arguments, Named Arguments, Scope of Variables, Lambda function.

Recursion Function Use of recursion function.

Unit III: Advanced Concepts

Functions, the Building Blocks of Code: Use of functions, Scopes and name resolution, input parameters, return values, Recursive functions, Anonymous functions, Function attributes, Built-in functions, importing objects. **Saving Time and Memory:** map, zip, and filter, Comprehensions, Generators, some performance considerations, Name localization, and Generation behavior in built-ins.

Advanced Concepts: OOP, Decorators, and Iterators: Decorators, Class and object namespaces, Attribute shadowing, initializing an instance, Accessing a base class, Multiple inheritance, Static and class methods, Private methods and name mangling, The property decorator, Operator overloading, Polymorphism

Unit IV: Web Development

The Edges – GUIs and Scripts: Scripting-The imports, Parsing Arguments, The business logic, GUI application- The import, The layout logic, The business logic, The tkinter.tixmodule, The turtle module, wxPython, PyQt, and PyGTK, The principle of least astonishment, Threading considerations. **Web Development Done Right:** Django design philosophy, The Django URL dispatcher, setting up Django, Adding the Entry model, Customizing the admin panel, Creating the form, Writing the views, tying up URLs and views, Writing the templates, writing a Flask view, Building a JSON quote server in Falcon.

Unit V: Cloud Native Python

Building Microservices in Python, Modeling microservices, building microservices, Testing the RESTful API. Building a Web Application in Python: Getting started with applications, working with Observables and AJAX, Binding data for the adduser template, working on Observables with AJAX for the addtweet template, Data binding for the addtweet template, CORS - Cross-Origin Resource Sharing, Session management, Cookies. **Interacting Data Services:** MongoDB terminology, Initializing the MongoDB database, integrating microservices with MongoDB, working with user resources, Working with the tweets resources.

Unit VI: Exception Handling

Testing, Profiling, and Dealing with Exceptions: The anatomy of a test, testing guidelines, Unit testing, Test-driven development, Exceptions, Profiling Python. Debugging and Troubleshooting: Debugging with print, Debugging with a custom function, Inspecting the traceback, Using the Python debugger, Inspecting log files, Other techniques,
Troubleshooting guidelines.

Text Books:

1. Learn Python Programming, 2nd Edition by Fabrizio Romano
2. Python Cookbook, 3rd Edition by David Beazley (Author), Brian K. Jones (Author)

Reference Books:

1. Python Programming: A Step-by-Step Guide For Absolute Beginners by Brian Jenkins and ATS Coding Academy
2. Python and AWS Cookbook: Managing Your Cloud with Python and Boto by Mitch Garnaat
3. Advanced Python Programming: Build high performance, concurrent, and multi-threaded apps with Python using proven design patterns by Dr. Gabriele Lanaro
4. Programming Google App Engine with Python: Build and Run Scalable Python Apps on Google's Infrastructure by Dan Sanderson

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the syntax, semantics, and basic constructs of Python programming, including variables, data types, and operators.
CO2: Apply control structures such as decision-making and loops to develop logic-based Python programs.
CO3: Develop modular programs using functions, built-in libraries, and user-defined modules.
CO4: Implement programs using data structures such as strings, lists, tuples, sets, and dictionaries.
CO5: Create file handling programs and apply object-oriented programming concepts such as classes, objects, inheritance, and polymorphism in Python.

BCAICC203: Power BI						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite:

1. Basic Knowledge of Excel and Data Handling
2. Fundamentals of Data Analysis and Visualization

Course Objective:

1. Understand data visualization principles and Power BI's role in business intelligence.
2. Learn data importing, cleaning, and transformation using Power Query.
3. Create interactive dashboards and reports for insightful data analysis.
4. Apply DAX functions for advanced calculations and data modeling.

UNIT I

Introduction to Power BI: Introduction to Power BI, Installation and setup of Power BI Desktop, Need for Power BI, Importance of Power BI, Introduction to Business Intelligence, Traditional BI vs. Power BI, Power BI vs Excel, Uses of Power BI.

UNIT II

Power BI Interface Overview: Importing and Connecting Data (Excel, CSV, databases, and cloud sources). Exploring the Fields, Visualizations, and Filters panes.

UNIT III

Data Transformation: Basic transformation, Introduction to Power Query, Use the first row as a header, Remove row, Add new columns, Add conditional columns, Remove blank/null values.

UNIT IV

Data Modeling: Understanding relationships between tables, Create relationships between multiple tables, Creating calculated columns, measures, and hierarchies, Introduction to DAX (Data Analysis Expressions),

UNIT V

Data Visualization: Types of visualizations: Tables, Charts, Maps, Cards, etc, Customizing visuals (filters, slicers,), Formatting and enhancing visual appeal, Creating dashboards with KPIs. Saving & Publishing your report

Textbooks:

1. "Mastering Microsoft Power BI" – Brett Powell, 2nd Edition (2022)
2. "Microsoft Power BI Quick Start Guide" – Devin Knight, Brian Knight, Mitchell Pearson, Manuel Quintana, 3rd Edition (2023)

Reference Books:

1. "Pro Power BI Desktop" – Adam Aspin, 2nd Edition (2020)
2. "Analyzing Data with Power BI and Power Pivot for Excel" – Alberto Ferrari & Marco Russo, 1st Edition (2017)

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the core concepts of business intelligence and the functionality of Power BI.
CO2: Import, clean, and transform data from various sources using Power BI tools.
CO3: Create data models and relationships to support efficient data analysis.
CO4: Design interactive dashboards and reports using various visualization techniques.
CO5: Apply DAX functions to perform complex calculations and enhance analytical capabilities.

BCAICC204: Operating Systems						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: - Programming languages, Data Structures, Microprocessor peripherals and interfacing

Course Objectives:

1. Define and list the functions of an operating system.
2. list resources involved in process creation and management.
3. Explain the use of paging and segmentation
4. Explain the function and structure of the I/O system.
5. Describe path names and directory structure visible to end users

Unit I: Introduction to Operating System
Introduction, Objectives and Functions of OS, Evolution of OS, OS Structures, OS Components, OS Services, System calls, System programs, Virtual Machines.

Unit II: Process Management
Processes: Process concept, Process scheduling, Co-operating processes, Operations on processes, Inter process communication, Communication in client-server systems. Threads: Introduction to Threads, Single and Multi-threaded processes and its benefits, User and Kernel threads, Multithreading models, threading issues.

Unit III: CPU Scheduling
CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling Algorithms, Multiple Processor Scheduling, Real-time Scheduling, Algorithm Evaluation, Process Scheduling Models. Process Synchronization: Mutual Exclusion, Critical – section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, Critical Regions, Monitors, OS Synchronization, Atomic Transactions Deadlocks: System Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Unit IV: Storage Management
Memory Management: Logical and physical Address Space, Swapping, Contiguous Memory Allocation, Paging, Segmentation with Paging. Virtual Management: Demand paging, Process creation, Page Replacement Algorithms, Allocation of Frames, Thrashing, Operating System Examples, Page size and other considerations, Demand segmentation File-System Interface: File concept, Access Methods, Directory structure, File- system Mounting, File sharing, Protection and consistency semantics.

Unit V: File-System Implementation:
File-System structure, File-System Implementations, Directory Implementation, Allocation Methods, Free-space Management, Efficiency and Performance, Recovery Disk Management: Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk Attachment, stable-storage Implementation

Unit VI: Protection and Security

Protection: Goals of Protection, Domain of Protection, Access Matrix, and Implementation of Access Matrix, Revocation of Access Rights, Capability- Based Systems, and Language – Based Protection. Security: Security Problem, User Authentication, One – Time Password, Program Threats, System Threats, Cryptography, Computer – Security Classifications.

.

Text Books:

1. Milan Milonkovic, Operating System Concepts and design, II Edition, McGraw Hill 1992.
2. Tanenbaum, Operation System Concepts, 2nd Edition, Pearson Education.
3. Silberschatz / Galvin / Gagne, Operating System,6th Edition,WSE (WILEY Publication)

Reference Books:

1. William Stallings, Operating System, 4th Edition, Pearson Education.
2. H.M.Deitel, Operating systems, 2nd Edition ,Pearson Education
3. Nutt: Operating Systems, 3/e Pearson Education 2004.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the basic concepts, functions, and types of operating systems, including system components and services.
CO2: Analyze process management techniques including process scheduling, creation, synchronization, and inter-process communication.
CO3: Apply concepts of memory management such as paging, segmentation, and virtual memory.
CO4: Understand and evaluate file systems, file access methods, and disk scheduling algorithms.
CO5: Demonstrate knowledge of deadlock detection, prevention, and recovery techniques in multi-process environments.

BCAICC205: Environmental Studies						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: - General knowledge of Ecology and Environment Biotechnology

Course Objectives:

1. Investigate the complexities of the natural environment and our relationship with it.
2. Explore the problems we face in understanding our natural environment and in living sustainability.
3. Develop scientific, interpretive and creative thinking skills.
4. Learn to apply quantitative analysis and field research techniques.
5. Use computer-based geographical information systems to study environmental change.

Detailed Syllabus

Unit I:
Introduction and natural resources: Multidisciplinary nature and public awareness, renewable and non-renewable resources and associated problems, forest, water, mineral, food, energy and land resources. Introduction to natural resources, conservation of natural resources and human role.

Unit II:
Ecosystem: Ecological concepts, concept of ecosystems, types of ecosystems, ecosystem structure and functioning, energy flow, food chains and food webs, ecological pyramids.

Unit III:
Biodiversity and Conservation: Definition, genetic species and ecosystem diversity biogeographically, classification of Indian value of biodiversity at national and local levels, India as a mega-diversity nation, treats to biodiversity and endangered and endemic species of India, need for conservation of biodiversity.

Unit IV:
Environmental pollution: Definition, causes, effect and control of air pollution, water pollution, soil pollution, marine pollution, noise pollution, thermal pollution, electromagnetic pollution, nuclear hazards , human role in prevention of pollution, solid waste management, disaster management, floods , earthquake, cyclone, and landslide
Firework Safety: Combustion of firework and pollution (noise, smoke, fireworks fallout and residue pollution), heavy metal toxicity due to fireworks and associated health effects.

Unit V:
Social Issue and Environment: Unsuitable to suitable development, urban problem related to energy and water conservation, environment protection act, wild life protection act, forest conservation act, environmental issues, population explosion, and family welfare programme. Environmental and human

health HIV, women and child welfare, role of information technology on environment and human health.
Corruption: definition and reasons, details of organizations/agencies working against corruption, role of individual against corruption and mode of action.
Ethics: Meaning, nature, determinants and objectives of ethics, ethics and its relation to values norms and morals, Indian ethos, Swami Vivekananda and ethics.

Text/Reference Books

1. Fundamentals of Environmental Biology, K. C. Agrawal, Nidhi Publishers (Bikaner).
2. Fundamentals of Ecology, E.P. Odum, W.B. Saunders Co. (USA).
3. Fundamentals of Ecology, E. P. Odum, Natraj Publisher (Dehradun).
4. Ecology: Principles and Applications, J. L. Chapman & , M. J. Reiss, Cambridge University Press.
5. Atmospheric pollution, W. Buch , Tata McGraw Hill(TMH)
6. Professional Ethics and Human Values, M. Govindarajan, PHI Learning Private Limited (Delhi).
7. Corruption and Reform in India, Jennifer Bussell , Cambridge University Press.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the structure and function of ecological systems and their relevance to environmental sustainability.
CO2: Analyze the causes and effects of environmental pollution and evaluate control measures to protect natural resources.
CO3: Demonstrate awareness of environmental policies, laws, and practices for sustainable development at local, national, and global levels.
CO4: Evaluate the role of biodiversity and conservation strategies in maintaining ecological balance.
CO5: Apply knowledge of renewable and non-renewable resources to suggest practical solutions for environmental challenges.

Syllabus

Semester: III

BCAICC301: Database Management System						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Computer Organization, Operating System, Data Structure, Mathematics

Course Objectives:

1. To introduce the fundamental concepts of database systems, data models, and database architecture.
2. To provide knowledge of relational databases, SQL, and techniques for effective data storage, retrieval, and manipulation.
3. To develop an understanding of database design using Entity-Relationship (ER) modeling and normalization techniques.
4. To enable students to implement and manage databases with attention to data integrity, security, and transaction management.

Detailed Syllabus:

Unit I: Introduction

Purpose of Database System -– Views of data – Data Models – Database Languages — Database System Architecture – Database users and Administrator – Entity– Relationship model (E-R model) – E-R Diagrams -- Introduction to relational databases

Unit II: Relational Model

The relational Model – The catalog- Types, Relational Algebra – Domain Relational Calculus – Tuple Relational Calculus - Fundamental operations – Additional Operations- SQL fundamentals, Oracle data types, Data Constraints, Column level & table Level Constraints, working with Tables, Defining different constraints on the table, Defining Integrity Constraints in the ALTER TABLE Command, Select Command, Logical Operator, Range Searching, Pattern Matching, Oracle Function, Grouping data from Tables in SQL, Manipulation Data in SQL.

Unit III: Relational Databases:

Relational Databases: Introduction to Relational Databases and Terminology-Relation, Tuple, Attribute, Cardinality, Degree, Domain. Keys, Super Key, Candidate Key, Primary Key, Foreign Key, Relational Algebra. Operations, Select, Project, Union, Difference, Intersection Cartesian product, Join, Natural Join.

Unit IV: SQL

Joining Multiple Tables (Equi Joins), Joining a Table to itself (self Joins), Sub queries Union, intersect & Minus Clause, creating view, Renaming the Column of a view, Granting Permissions, - Updating, Selection, destroying view Creating Indexes, Creating and managing User, Integrity – Triggers - Security – Advanced SQL features –Embedded SQL– Dynamic SQL- Missing Information– Views – Introduction to Distributed Databases and Client/Server Databases

Unit V: Database Design
Functional Dependencies – Non-loss Decomposition – Functional Dependencies – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Normal Form-Multi-Valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form

Unit VI: Transactions
Transaction Concepts - Transaction Recovery – ACID Properties – System Recovery – Media Recovery – Two Phase Commit - Save Points – SQL Facilities for recovery –Concurrency – Need for Concurrency – Locking Protocols – Two Phase Locking – Intent Locking – Deadlock-Serializability – Recovery Isolation Levels – SQL Facilities for Concurrency.

Text Book:
1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Fifth Edition, Tata McGraw Hill, 2006.
2. RamezElmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Fourth Edition, Pearson/Addision Wesley, 2007.

Reference Books:
1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Fifth Edition, Tata McGraw Hill, 2006
2. RamezElmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Fourth Edition, Pearson/Addision Wesley, 2007.
3. Raghu Ramakrishnan, "Database Management Systems", Third Edition, McGraw Hill, 2003.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts of database systems, data models, and database architecture to design efficient data storage solutions.
CO2: Apply Entity-Relationship (ER) modeling to design relational databases and convert ER diagrams into normalized relational schemas.
CO3: Construct and execute queries using Structured Query Language (SQL) to manipulate and retrieve data effectively.
CO4: Analyze and apply normalization techniques and functional dependencies to optimize database design and reduce data redundancy.
CO5: Understand and implement concepts of transaction management, concurrency control, and database recovery to ensure data integrity and consistency.

BCAICC302: Data Structures						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Programming languages, Data Structures, Microprocessor peripherals and interfacing.

Course Objectives:

1. Define and list the functions of an operating system.
2. list resources involved in process creation and management.
3. Explain the use of paging and segmentation
4. Explain the function and structure of the I/O system.
5. Describe path names and directory structure visible to end users

Detailed Syllabus:

Unit I: Introduction to Data structures

Introduction: Definition, Classification of data structures: primitive and non-primitive, Elementary data organization, Time and space complexity of an algorithm (Examples), String processing. Dynamic memory allocation and pointers: Definition of dynamic memory allocation, Accessing the address of a variable, Declaring and initializing pointers, Accessing a variable through its pointer, Meaning of static and dynamic memory allocation, Memory allocation functions: malloc(), calloc(), free() and realloc(). Recursion: Definition, Recursion in C (advantages), Writing Recursive programs – Binomial coefficient, Fibonacci, GCD.

Unit II: Searching and Sorting

Basic Search Techniques: Sequential search: Iterative and Recursive methods, Binary search: Iterative and Recursive methods, Comparison between sequential and binary search. Sort: General background and definition, Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort.

Unit III: Stack and Queue

Stack: Definition, Array representation of stack, Operations on stack: Infix, prefix and postfix notations, Conversion of an arithmetic expression from Infix to postfix, Applications of stacks.

Queue: Definition, Array representation of queue, Types of queue: Simple queue, Circular queue, Double ended queue (deque), Priority queue, Operations on all types of Queues.

Unit IV: Linked List

Linked List: Definition, Components of linked list, Representation of linked list, Advantages and Disadvantages of linked list. Types of linked list: Singly linked list, doubly linked list, Circular linked list, Operations on singly linked list: creation, insertion, deletion, search and display.

Unit V: Tree Graphs and their Applications:

Definition: Tree, Binary tree, Complete binary tree, Binary search tree, Heap Tree

Terminology: Root, Node, Degree of a node and tree, Terminal nodes, Non-terminal nodes, Siblings, Level, Edge, Path, depth, Parent node, ancestors of a node. Binary tree: Array representation of tree,

Creation of binary tree. Traversal of Binary Tree: Preorder, Inorder and postorder. Graphs, Application of Graphs, Depth First search, Breadth First search.

Unit V: Memory Management and File System:

Memory Management: Logical versus Physical Address Space, Swapping, Contiguous Allocation (Memory Allocation, Fragmentation), Paging (Basic Method, Hardware Support), Segmentation (Basic Method, Hardware). Virtual Memory: Demand Paging, Page Replacement, Page Replacement Algorithms.

File System: File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, Free-Space Management, Protection of File System. Input/output Management. Linux Case Study.

Text Book

1. Weiss, Data Structures and Algorithm Analysis in C, II Edition, Pearson Education, 2001
2. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill
3. Robert Kruse Data Structures and program designing using 'C'

Reference Books:

1. Trembley and Sorenson Data Structures
2. E. Balaguruswamy Programming in ANSI C.
3. Bandyopadhyay, Data Structures Using C Pearson Education, 1999
4. Tenenbaum, Data Structures Using C. Pearson Education, 200
5. Kamthane: Introduction to Data Structures in C. Pearson Education 2005.
6. Hanumanthappa M., Practical approach to Data Structures, Laxmi Publications, Fire Wall media 2006
7. Langsam, AusensteinMaoshe& M. Tanenbaum Aaron Data Structures using C and C++ Pearson Education

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts of data structures and their importance in algorithm design and software development.
CO2: Implement linear data structures such as arrays, linked lists, stacks, and queues for various computational problems.
CO3: Apply non-linear data structures like trees and graphs to solve hierarchical and network-based problems.
CO4: Analyze and compare the efficiency of different sorting and searching algorithms in terms of time and space complexity.
CO5: Choose appropriate data structures for problem-solving and develop efficient solutions using structured and modular programming techniques.

BCAICC303: Data Communication and Computer Networks						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: -

1. Familiarity with the fundamentals of Digital Electronics.
2. A network simulation method.

Course Objectives:

1. Learn how computer network hardware and software operate.
2. Investigate the fundamental issues driving network design.
3. Learn about dominant network technologies.

Detailed Syllabus

Unit I: Networking Fundamentals

Basics of Network & Networking, Advantages of Networking, Types of Networks, Types of Network Architecture, Workgroup vs. Domain. Network Topologies, Types of Topologies, Logical and physical topologies, selecting the Right Topology, Types of Transmission Media, Communication Modes, Wiring Standards and Cabling, media connectors, Introduction of OSI model, Functions of the seven layers, Introduction of TCP/IP Model, Comparison between OSI model & TCP/IP model.

Unit II: Basics of Network Devices

Network Devices- NIC- functions of NIC, installing NIC, Hub, Switch, Bridge, Router, Gateways, And Other Networking Devices, Repeater, CSU/DSU, Modem, Ethernet standards, Ethernet Components, Point-to-Point Protocol, Address Resolution Protocol, Message format, transactions, Benefits of Wireless Technology, Types of Wireless Networks, Wireless network Components, wireless LAN standards, wireless security Protocols.

Unit III: Basics of Network, Transport and Application Layers

Network Layer: Internet Protocol (IP), IP standards, versions, functions, The IPv4 and IPv6 Datagram Format, IPv4 addressing, IPv4 Subnetting, CIDR and VLSM, IPv6 Addressing, , Internet Control Message Protocol , Internet Group Management Protocol ,Introduction to Routing and Switching concepts, Transport Layer: Transmission Control Protocol(TCP), User Datagram Protocol (UDP), Overview of Ports & Sockets, Application Layer Protocols

Unit IV: WAN Technology

Introduction to WAN, WAN Switching techniques, connecting to the Internet, Satellite-Based Services, Cellular Technologies, Technologies used for Connecting LANs, Remote Access Connections and technologies, Authentication and Authorization, Tunnelling and Encryption Protocols, Security Appliances and Security Threats.

Unit V: Troubleshooting Network

Trouble Shooting Networks: Command-Line Interface Tools, Network and Internet Troubleshooting, Troubleshooting Model, identify the affected area, probable cause, implement a solution, test the result, recognize the potential effects of the solution, document the solution, Using Network Utilities: ping, traceroute, tracert, ipconfig, arp, nslookup, netstat, nbtstat, Hardware trouble shooting tools, system monitoring tools.

Unit V: Layers:

Network Layer: Point-to Point networks, Routing algorithms, Congestion control algorithms, Internetworking Devices, IP protocol, IP addresses: IPv4 classful and classless addressing, Introduction to IPv6

Transport Layer: Connection management: Three-way Handshaking. Introduction of User Datagram Protocol (UDP), Basics of Transmission Control Protocol. (TCP).

Application Layer: File Transfer Protocol, Domain Name System, Electronic mail, Intro of Client server model, Hyper Text Transfer Protocol, WWW, Example Networks - Internet and Public Networks

Text Book:

1. CCNA Cisco Certified Network Associate: Study Guide (With CD) 7th Edition (Paperback), Wiley India, 2011
2. CCENT/CCNA ICND1 640-822 Official Cert Guide 3 Edition (Paperback), Pearson, 2013

Reference Books:

1. Routing Protocols and Concepts CCNA Exploration Companion Guide (With CD) (Paperback), Pearson, 2008
2. CCNA Exploration Course Booklet: Routing Protocols and Concepts, Version 4.0 (Paperback), Pearson, 2010

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the basic concepts of data communication, network models, and transmission media used in modern communication systems.
CO2: Explain the functions and protocols of different layers in the OSI and TCP/IP network models.
CO3: Analyze error detection and correction techniques, as well as flow and congestion control mechanisms in reliable data transfer.
CO4: Evaluate different switching techniques, IP addressing schemes, and routing algorithms used in network communication.
CO5: Demonstrate knowledge of various network topologies, protocols, and security measures used in wired and wireless networks.

BCAI304: Statistics and Probability						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Algebra and Arithmetic Operation, Familiarity with Data Representation and Interpretation

Course Objectives:

1. Understand the fundamental concepts of descriptive and inferential statistics.
2. Develop the ability to compute and interpret measures of central tendency, dispersion, and correlation.
3. Gain knowledge of probability theory and its applications in real-world scenarios.
4. Apply statistical methods for data analysis and decision-making under uncertainty.

Detailed Syllabus

Unit I: Descriptive Statistics

Data and Data Sources, Types of Data, Measures of Central Tendency – Mean, median mode for raw and grouped data, measures of dispersion – Range, standard deviation, variance, coefficient of variation, mean deviation, mean absolute deviation, measures of symmetry: Skewness and Kurtosis.

Unit II: Elements of Probability and Sampling Distributions

Experiments and events, Basic Relations of Probability, Conditional Probability, Joint Probability, conditional probability on discrete case and continuous case, computing expectations by conditioning, introduction to Bayes theorem, problems related to Bayes Theorem, Discrete Probability Distribution (Binomial and Poisson), Continuous Probability Distribution (Normal). Various types of Probability and Non-Probability Sampling, Sampling distribution of important statistic.

Unit III: Hypothesis Testing

Introduction, Statistical assumptions for parametric test, Level of significance, confidence level, Type I Error, Type II error, Critical value, power of the test, sampling distribution, small sample test – t test, F test, Large Sample test – Z test for equality of single mean with population mean, equality of two sample mean, equality of single proportion with population proportion and equality of two sample proportions.

Unit IV: Correlation and Regression Analysis

Correlation analysis, properties, significance of single and multiple correlation coefficients, multiple and partial correlation, linear model, factor, effect, residuals, dependency, independency, assumptions of linear model, estimation of parameters of regression coefficients, properties, significance, diagnostic testing: auto correlation, multi collinearity, heteroscedasticity, normality, significance of estimated parameters in multiple linear regression,.

Unit V: Linear Model

Introduction, assumptions, factors and levels in ANOVA, layout of one way ANOVA, multiple comparison of sample means, one way analysis of variance with unequal sample sizes, two factor analysis of variance – introduction and parameter estimation, two way analysis of variance with

interaction, Post ANOVA: testing of hypothesis for significance of mean using Fishers Least Significance Difference test (lsd), Tukeys test, Dunnet test, Duncan Multiple Range test.

Text Book:

1. Fundamentals of mathematical statistics – SC Gupta and VK Kapoor, Sultan Chand & Sons Publication, New Delhi

Reference Books:

1. Introduction to probability Models, Ninth Edition – Sheldon M. Ross, Elsevier Publication, Academic Press, UK
2. Introduction to Probability and Statistics for Engineers and Scientists, Third Edition - Sheldon M. Ross, Elsevier Publication, Academic Press, UK

Course Outcomes:

On successful completion of the module students will be able to:

CO1: Interpret and summarize data using descriptive statistical methods.
CO2: Apply probability concepts to analyze random events and outcomes.
CO3: Use statistical tools to estimate population parameters and test hypotheses.
CO4: Solve problems involving discrete and continuous probability distributions.
CO5: Analyze relationships between variables using correlation and regression techniques.

BCAI305 (Elective-I): Digital Electronics						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic knowledge of Computers Fundamentals and Physics of Intermediate standard.

Course Objectives:

1. To describe various types of Number System, basic electronic components and hardware components of computer system.
2. To understand the concept of Boolean algebra, types of digital circuits, memories, addressing modes and I/O interface.
3. To solve problems related to number system conversions and calculation of binary codes.
4. To implement basic Boolean expressions using different Digital Electronic device.
5. To distinguish between types of digital circuits, addressing modes, memories and I/O interface.
6. To design digital circuits for a particular functions using basic electronic concept.

Detailed Syllabus

Unit I: Basic Concepts, Boolean algebra, Theorems and Functions
Number Systems: Decimal number system, binary number system, octal number system, hexadecimal number system, BCD number system, Excess-3 code, Gray code, Alpha numeric code, error detecting and error correcting codes. Arithmetic: Arithmetic number representation, Binary arithmetic, Hexadecimal arithmetic, BCD arithmetic. Boolean algebra and Theorems: Logic gates and logic operations, Boolean theorems and postulates, SOP's & POS's, Minterms and Maxterms. Minimization of Boolean Functions: Algebraic simplification, Karnaugh map simplification, Quine-Mc Cluskey or Tabulation method.

Unit II: Logic Gates
Logic Families: Metal Oxide Semiconductor logic families- switching properties of NMOS and PMOS transistors, static NMOS, dynamic NMOS, Static CMOS and dynamic CMOS logic families, CMOS Transmission gate circuits, Bipolar logic families- switching properties of NPN and PNP transistors, TTL, Schottkey TTL, Comparison of MOS logic circuits(CMOS) with that of a TTL digital circuit, Tristate gates. Electrical characteristics: Meanings of speed, propagation delay, operating frequency, and power dissAIted per gate, supply voltage levels, operational voltage levels of various logic families.

Unit III: Combinational Systems
Binary arithmetic units (Adder, Subtractor, n-bit parallel adder & Subtractor, look ahead carry generator), decoder, encoder, multiplexer, Demultiplexer, code converters, Magnitude comparators, parity generators. Implementation of combinational logic by standard IC's.

Unit IV: Sequential Systems
Flip-flop and Latch: SR latch, JK flip-flop, T flip-flop, D flip-flop and latch, Master-slave RS flip-flop, Master-slave JK flip-flop, asynchronous inputs. Registers & Counters: Shift registers (SISO, SIPO, PISO, PIPO), universal shift register. Counters Asynchronous/Ripple counters, Synchronous counters,

Modulus-n Counter, Ring counter, Johnson counter, Up-Down counter, asynchronous clear, preset and load in a counter, synchronous clear, preset and load in a counter, typical IC's for counters. Synchronous (Clocked) sequential circuits: Moore and Mealey state machine circuits, Analysis & design of synchronous sequential circuits – State machine design with SM charts

Unit V: Memory and Programmable Logic
RAM, memory decoding, ROM, PROMs, PAL & PLA, Sequential Programmable Devices (discuss three major devices without going into their detailed construction).

Unit V: Processor Organization and I/O Organization:
Processor Organization- Introduction and types of CPU Organization, addressing modes, Implied Addressing Mode, Immediate Addressing Mode, Register Addressing Mode, register indirect Addressing Mode, Direct Addressing Mode, Indirect Addressing Mode, Relative Addressing Mode, index Addressing Mode, auto increment/decrement Addressing Modes.
I/O Organization - Introduction to I/O organization, I/O interface and its need.
Memory Organization- Memory Hierarchy, RAM and ROM chips, SRAM, DRAM, PROM, EEPROM, Introduction of Cache Memory and Virtual Memory.

Text Book:

1. Morris Mano M, Michael D. Ciletti, "Digital Design", Pearson Education, 4th Edition, 2007.
2. Charles H Roth (Jr), Larry L. Kinney, "Fundamentals of Logic Design", Cengage Learning India Edition, 5th Edition, 2010.
3. Floyd and Jain, "Digital Fundamentals", Pearson Education, 8th Edition, 2007.

Reference Book:

1. Ronald J. Tocci, "Digital Systems: Principles and Applications", Pearson Education, 10th Edition, 2009.
2. Donald P Leach, Albert Paul Malvino, Goutam Saha, "Digital Principles and Applications", Tata McGraw Hill, 6th Edition, 2008

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental principles of number systems, binary arithmetic, and logic gates used in digital circuits.
CO2: Design and analyze combinational circuits such as adders, multiplexers, encoders, and decoders.
CO3: Explain the operation and applications of sequential circuits including flip-flops, counters, and shift registers.
CO4: Understand the basic structure, functioning, and organization of computer systems, including CPU, memory, and input-output devices.
CO5: Analyze instruction formats, addressing modes, and assembly-level programming concepts in computer architecture.

BCAI306 (Elective-I): Data Visualization						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Understanding of Statistics and Data Analysis, Familiarity with Data Handling Tools (e.g., Excel, Python, or R)

Course Objectives:

1. Understand the principles and importance of effective data visualization.
2. Learn to use tools and libraries (e.g., Tableau, Matplotlib, Power BI) for creating visual representations of data.
3. Develop the ability to choose appropriate charts and graphs based on data types and analysis goals.
4. Apply visualization techniques to communicate insights and support data-driven decision-making.

Detailed Syllabus

Unit I: Introduction to Data Visualization

Brief history of data visualization, scientific design choices in data visualization- choice of graphical form, grammar of graphical techniques of large amount of data, crucial need of visualization techniques, challenges in visualization techniques, classification of visualization techniques for qualitative and quantitative data, power of visualization techniques, introduction to different visualization techniques.

Unit II: Static Graphical Techniques

Introduction to bar graph, basic understanding of making basic bar graph, grouping bars together, bar graphs on counts, customization of bar graphs by changing color, size, title, axis units, changing width and spacing of the bar chart, adding labels to bar graph, application of bar graph in business.

Unit III: Multivariate Graphical Techniques

Introduction to correlation matrix, application of correlation matrix in the multivariate analysis, network graph, basics of heat map, difference between heat map and tree map, introduction to higher dimensional scatter plot, axis adjustment in the higher dimensional scatter plot, addition of prediction surface of higher dimensional scatter plot.

Unit IV: Graphical Validation

Basics of multivariate statistical visual representations and its results, dendrogram, importance of dendrogram in grouping (cluster analysis), Scree Plot, importance of Scree Plot, application of Scree Plot in determining number of clusters and factors.

Unit V: QQ and PP ploting:

QQ plot, importance of QQ plot in distribution of data for the further quantitative analysis, PP plot, applications and usage of PP Plot for distribution detection.

Unit VI: Customization

Introduction to annotations – adding : text, mathematical expression , lines, arrows, shaded shapes, highlighting the texts and items, adding error bars, introduction to axis, swapping x and y axis, changing the scaling ration in the axis, positioning of axis and arranging tick marks and labels, changing the appearance of axis labels, circular graphs, using themes, changing the appearance of theme elements, creating the own themes, legends : removing the legends, position of legends, legend title, labels in legends.

Text Books

1. Data Visualization Principles And Practice, Second Edition - Alexandru Telea, Crc Press.
2. Hand book of Data Visualization – Chun-houh Chen, Wolfgang Härdle, Antony Unwin, Springer Publication.

Reference Books

1. R Graphics Cook Book, Winston Chang, First Edition, O'Reilly Publication.
2. ggplot2 Elegant Graphics for Data Analysis – Hadley Wickham, Springer Publication

Course Outcome:

At the end of this course students able to

CO1: Identify and apply suitable visualization techniques for various data types.
CO2: Use data visualization tools and libraries to create insightful visualizations.
CO3: Interpret and communicate data-driven insights effectively through visual means.
CO4: Design interactive dashboards and reports for real-time data analysis.
CO5: Evaluate and improve visualizations based on clarity, accuracy, and audience understanding.

Syllabus

Semester: IV

BCAI401: Introduction to Artificial Intelligence						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: - Should have knowledge about advanced mathematics

Course Objectives:

1. The main objective of AI to build intelligent machine which can perform and act like humans.
2. The main objective of this course is to understand how these algorithms works and how to analyze the data to make a proper decision.
3. As we know AI is in used in all fields like healthcare industry, mobile world, Retail, Fraud detection etc. so demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
4. To initiate the concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems in different fields.
5. To evaluate the different stages of development of the AI field from human like behavior to Intelligent Agents.

Detailed Syllabus

Unit I: Cognitive Process Automation concepts:
Introduction to CPA: Scopes and techniques of CPA, CPA features, CPA platform overview, The future of intelligent automation. Exploration of the tool: UAIth architecture, Installing and Learning UAIth studio, UAIth operating model, Database installation.

Unit II: Automation in UAIth
UAIth: Working with different stages, Calculation, Decision, Choice, Collection, Loop, Anchor, Understanding Business objects, Understanding UAIth processes, Pages, Multi Page and page linking, Input, Output and Startup Parameters. End to End Automation: Creating and Managing Business objects in object studio, Creating and Managing UAIth processes in process studio, CSV/Excel to data table transfer and vice versa.

Unit III: UAIth Life Cycle and their artifacts
User Interface Components: Ribbon, Toolbars Access, Library panel, project panel, Outline panel, locals panel, Debugging, Recording, Workflow execution, context menu, properties panel, Designer panel, Universal search bar. UI Automation and System Activities: UI automation, System, Properties, Variables, Output and Arguments.

Unit IV: Natural Language Processing:
Text Analysis, Text Cleaning, Stemming, TDM and DTM, Sentiment Analysis, NLP API consumption, Build your own social media monitoring tool and Analysis of Email. Chatbot: Handling user events and assistant Bots, Monitoring system event triggers, Hotkey triggers, Mouse triggers, System triggers, Launching an assistant bot on a keyboard event.

Unit V: Image and Text Automation:
Image Automation: Mouse and keyboard activitites, Guides/text activities, OCR- activities, Types of OCR, Image Activities, Computer Vision, Image classification, Unstructured data to structure conversion, Invoice data extraction. Text Automation: Exception Handling, Logging, Debugging, Tracing, Connecting with Database, Executing Query with Database, Project Organization, PDF-data extraction and automation, Email automation.

Unit V: Probabilistic Reasoning and Expert System:
Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Dempster-Shafer Theory. Introduction of Natural Language Processing
Expert System: Rule based system architecture, Non production system architecture, knowledge organization and validation, Existing Systems (DENDRAL, MYCIN).

Text Books:
1. Robotic Process Automation Tools, Process Automation and their benefits: Understanding RPA and Intelligent Automation by Mr Srikanth Merianda.
2. Intelligenct Control: A stochastic optimization approach by Kaushik Das Sharma, Amitava Chatterjee, Anjan Rakshit. –Springer edition.
3. Learning Robotic Process Automation by Alok Mani TrAIthi.

Reference Books:
1. Robotic Process Automation- Guide to building robots by Richard Murdoch.
2. Robotic Process Automation and Risk Mitigation: The Definitive Guide by Mary C. Lacity and Dr. Leslie P. Willcocks
3. Introduction to robotic process automation by Frank Casale

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts, history, and applications of Artificial Intelligence.
CO2: Apply search algorithms and problem-solving strategies in AI environments.
CO3: Implement knowledge representation techniques such as semantic networks, frames, and logic-based models.
CO4: Analyze and apply basic machine learning algorithms for classification and prediction.
CO5: Demonstrate the ability to design intelligent agents and solve real-world problems using AI techniques.

BCAI402: Sensor Technology						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Physics and Electronics, Understanding of Electrical Circuits and Signal Processing

Course Objectives:

1. Understand the fundamental principles and working mechanisms of various sensors.
2. Explore different types of sensors used in industrial, medical, and environmental applications.
3. Learn techniques for sensor interfacing, signal conditioning, and data acquisition.
4. Develop the ability to select appropriate sensors based on application-specific requirements.

Detailed Syllabus

Unit I: Science of Measurement and Instrumentation

Functional Elements of Measurement Systems. Definition, principles of sensing and transduction, Classification of transducers, Units and standards, Classification of errors Odds and uncertainty - Introduction to Calibration methods.

Unit II: Resistive Transducers

The principle of operation, construction details, characteristics and application of resistance potentiometer, strain gauge and its signal conditioning circuits, RTD, LDR, thermistor, hot-wire anemometer and humidity sensor. Demonstration of RTD, Strain gauge, LDR

Unit III: Inductive and Capacitive Transducers

Induction potentiometer - Variable reluctance transducers, LVDT - Variable Reluctance Tachometer, Proximity transducers - Capacitive transducer and types - Capacitor microphone

Unit IV: Other Transducers

A piezoelectric transducer, magnetostrictive transducer - Digital transducers - Fiber optic transducer - Hall Effect transducer - Photo electric transducer- I/P & P/I transducer. Introduction to Smart sensors and MEMS. Points to be considered for selecting a transducer.

Unit V: Sensor Applications

Typical applications of sensors, Sensors in Real time industry, Selection criterial of the sensor as per the application, Application Cases: Weather monitoring system, Water monitoring system, Sensor on the Conveyor system.

Text Book:

1. E.A. Doebelin, “Measurement Systems – Applications and Design”, Tata Mc Graw Hill, New York, 2012
2. John P. Bentley, “Principles of Measurement Systems”, 4th Edition, Pearson Education, 2005
3. S. Ranganathan, “Transducer Engineering”, Allied Publishers Pvt. Ltd.,2003

Reference Book:

1. D.Patranabis, “Sensors and Transducers”, Prentice Hall of India, 2004.

Course Outcome:

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

CO1: Describe the operating principles and characteristics of various types of sensors.
CO2: Select suitable sensors for specific industrial or environmental applications.
CO3: Demonstrate knowledge of sensor interfacing and signal conditioning techniques.
CO4: Analyze sensor data for monitoring and control purposes.
CO5: Design and implement basic sensor-based systems for real-world applications.

BCAI403 (Elective-II): Business Process Management						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Understanding of Business Operations and Organizational Structure, Familiarity with Information Systems and Workflow Concepts

Course Objectives:

1. To understand the fundamental concepts, lifecycle, and significance of business process management in organizations.
2. To analyze and model business processes using appropriate tools and techniques.
3. To learn strategies for process improvement, automation, and optimization.
4. To apply BPM principles in aligning business processes with organizational goals and enhancing overall performance.

Detailed Syllabus

Unit I: Introduction to Business Process Management

Introduction, definition, business process life cycle, classification of business process, goals, structures and organization, evolution of enterprise systems architecture, business process modelling foundation: Conceptual Model and Terminology, Activity Models and Activity Instances, Process Models and Process Instances, Process Interactions, Modelling Process Data, modelling organization and operations, business process flexibility.

Unit II: Process Orchestrations and Choreographies

Introduction, Control Flow Patterns, Petri Nets, Event-driven Process Chains, Workflow Nets, Yet Another and Graph Based Workflow Language, Business Process Modeling Notation, Development Phases, Process Choreography Design and implementation, service interaction patterns.

Unit III: Properties and Architecture of Business Process Management

Introduction, Data Dependencies, Structural Soundness, Soundness, Relaxed Soundness, Weak Soundness, Lazy Soundness, Workflow Management Architectures, Flexible Workflow Management, Web Services and their Composition, Advanced Service Composition.

Unit IV: Business Process Methodology – I

Introduction, Methodology Overview, Strategy and Organization, Survey, Design Phase, Platform Selection, Implementation and Testing, Operation and Controlling Phase.

Unit V : Business Process Methodology - II

Business Process Reengineering, Business Process Management (BPM) Implementation Methodology, Phases for Implementation: Commit, Research, Analyze, Design and Implement and Support.

Text Books:

1. Business Process Management - Concepts, Languages, Architectures by Mathias Weske, Springer, 2007.
2. BUSINESS PROCESS MANAGEMENT SYSTEMS by James F. Chang, Auerbach Publications, 2006.

Reference Books:

1. Business Process Management Practical Guidelines to Successful Implementations by John Jeston and Johan Nelis, Butterworth-Heinemann is an imprint of Elsevier, 2008
2. BUSINESS PROCESS MANAGEMENT APPLIED by Charles Poirier Ian Walker, J.Ross Publishing Inc, 2005

Course Outcome:

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

CO1: Understand and explain the principles, lifecycle, and importance of BPM in modern enterprises.
CO2: Model business processes using standard notations such as BPMN (Business Process Model and Notation).
CO3: Analyze and identify areas for improvement in existing business processes.
CO4: Apply techniques for process automation and optimization to improve efficiency and performance.
CO5: Evaluate BPM tools and technologies to support organizational goals and decision-making.

BCAI404 (Elective-II): Introduction to Data Science						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite:

1. Basic Knowledge of Programming and Data Structures
2. Understanding of Statistics and Mathematics

Course Objectives:

1. To introduce the fundamental concepts, tools, and techniques used in data science.
2. To develop skills in data collection, cleaning, visualization, and exploratory data analysis.
3. To enable students to apply statistical and machine learning techniques for data-driven decision-making.
4. To provide hands-on experience with programming tools such as Python and libraries like Pandas, NumPy, and Matplotlib.

Detailed Syllabus:

Unit I: Data Science - An Overview

Introduction to Data Science, Definition and description, history and development, terminologies, basic framework and architecture, difference between Data Science and business analytics, importance of Data Science, primary components of Data Science, users of Data Science and its hierarchy, overview of Data Science techniques, challenges and opportunities in Data Science, industrial application of Data Science techniques.

Unit II: Mathematics and Statistics in Data Science

Role of mathematics, importance of probability and statistics, important types of statistical measures: Descriptive, Predictive and prescriptive statistics, introduction to statistical inference, application of statistical techniques, linear algebra: matrix and vector theory, role of linear algebra in Data Science, exploratory data analysis and visualization techniques, difference between exploratory and descriptive statistics.

Unit III: Machine Learning in Data Science

Role of machine learning, different types of machine learning techniques and its broad scope: Supervised, unsupervised, reinforcement and deep learning, difference between different machine learning techniques, machine learning algorithms, importance of machine learning in today's business, difference between classification and prediction.

Unit IV: Computers in Data Science

Role of computer science in Data Science, various components of computer science being used for Data Science, role of relation data base systems: SQL, NoSQL, data warehousing, importance of operating concepts and memory management, freely available software tools: R, Python, important proprietary software tools, business intelligence tools.

Unit V: Data Science Project Management

Data Science project framework, execution flow of a Data Science project, various components of Data Science projects, stakeholders of Data Science project, industry use cases of Data Science implementation, challenges and scope of Data Science project management, process evaluation model, comparison of Data Science project methods, improvement in success of Data Science project models.

Text Books:

1. Data Science from Scratch: First Principles with Python 1st Edition by Joel Grus
2. Principles of Data Science by Sinan Ozdemir, (2016) PACKT.

Reference Book:

1. Data Science For Dummies by Lillian Pierson (2015)
2. Data Science for Business: What You Need to Know about Data Mining and Data- Analytic Thinking by Foster Provost, Tom Fawcett
3. Data Smart: Using Data Science to Transform Information into Insight 1st Edition by John W. Foreman. (2015) Wiley Publication.

Course Outcome:

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

CO1: Understand the fundamental concepts, tools, and techniques used in data science.
CO2: Collect, clean, and preprocess real-world datasets for effective analysis.
CO3: Apply statistical methods and machine learning algorithms to extract meaningful insights from data.
CO4: Visualize data using appropriate tools and libraries such as Matplotlib and Seaborn.
CO5: Develop data-driven solutions using Python and popular data science libraries like Pandas, NumPy, and Scikit-learn.

BCAI405 (Elective-II): Pattern Recognition						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Linear Algebra, Probability, and Statistics, Understanding of Programming Languages (e.g., Python, MATLAB) and Machine Learning Fundamentals

Course Objectives:

1. To understand the fundamental concepts, models, and techniques used in pattern recognition.
2. To explore statistical, structural, and neural approaches for pattern classification and clustering.
3. To develop the ability to apply pattern recognition algorithms to real-world data sets.
4. To analyze the performance of classification systems using appropriate evaluation metrics.

Detailed Syllabus

Unit I: Introduction to Pattern Recognition Systems

The Design Cycle-Learning And Adaptation-Supervised Learning- Unsupervised Learning-Reinforcement Learning-Linear Model For Classification-Discriminant Function(Two Class And Multiclass)-Least Square For Classification- Fisher's Linear Discriminant Analysis For Two And Multiple Class- Probabilistic Generative Models – Maximum Likelihood Solution.

Unit II: Parameter Estimation and Supervised Learning

Maximum Likelihood Estimation - The Bayes Classifier - Learning The Mean Of A Normal Density - General Bayesian Learning - Nonparametric Technic – Density Estimation - Parzen Windows - K-Nearest Neighbor Estimation - Estimation Of Posterior Probabilities - Nearest- Neighbor Rule - K-Nearest Neighbor Rule.

Unit III: Kernel Methods

Constructing Kernels – Kernel Density Estimators - Nearest Neighbor Methods – Gaussian Processes And Classification – Sparse Kernel Machines - Support Vector Machines – Maximum Margin Classifiers - Multi-Class Support Vector Machine. Graphical Models: Bayesian Networks - Generative Models - Linear Gaussian Models - Conditional Independence.

Unit IV: Gaussian Mixture Models and Expectation Maximization

K-Means Clustering - Mixtures Of Gaussian -Expectation Maximum For Gaussian Mixtures. Continuous Latent Variables: PrincAIl Component Analysis -Applications Of PrincAIl Component Analysis -PCA For Higher Dimensional Data - Factor Analysis. Sequential Data: Markov Models – Hidden Markov Models - Maximum Likelihood For HMM–Forward- Backward Algorithm. Combining Models.

Unit V: Tree Based Models

Tree based model- Pros and Cons, decision tree algorithms, classification tree, characteristics of classification tree, induction algorithms, probability estimation in decision tree – Laplace correction and no match method, stopping criteria for tree development, pruning techniques and pruned tree.

Classification Methods: Information Gain: ID3-C4.5-C5-J 48. Gini Index: SPRINT-SLIQ. Decision Tree Advantages and Disadvantages-Application of decision tree- Classification and Regression Trees (CART).

Text Books:

1. "Pattern Recognition And Machine Learning",. Christopher M. Bishop , Springer, 2006
2. "Pattern Classification", Second Edition, Richard O. Duda, Peter E. Hart And David G. Stork John Wiley & Sons, 2003

Reference Books:

1. Syntactic Pattern Recognition And Applications, Fu K.S., Prentice Hall, Eaglewood Cliffs
2. Pattern Recognition: Techniques And Applications by RajjanShinghal : Oxford University Press, 2008,
3. Pattern Classification And Scene Analysis, John Wiley, Duda& Hart P.E.
4. Syntactic Pattern Recognition - An Introduction by Addison Wesley Gonzalez R.C. & Thomson M.G.

Course Outcome:

After completing the course, students will be able to:

CO1: Understand the foundational principles and methodologies of pattern recognition.
CO2: Apply statistical and machine learning techniques to classify and cluster data.
CO3: Design and implement pattern recognition systems using real-world datasets.
CO4: Analyze and evaluate the performance of pattern recognition algorithms.
CO5: Demonstrate the ability to choose appropriate models and tools for specific pattern recognition tasks.

BCAICC406: Responsive Web Design-Front End Development						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: HTML and CSS.

Course Objectives:

1. Understand and Apply the Fundamental Concepts of HTML, CSS, and JavaScript to Build Structured, Responsive, and Interactive Web Pages
2. Evaluate understanding of responsive design principles, advanced CSS techniques, and the use of preprocessors.
3. Design and Implement Responsive Web Interfaces Using Advanced CSS Techniques and Preprocessors to Enhance User Experience Across Devices
4. Write simple JavaScript code to manipulate the DOM and respond to user interactions.
5. Analyze and Implement Advanced JavaScript Techniques to Develop Modular, Efficient, and Maintainable Web Applications.

Module I: Introduction to HTML

What is HTML: Definition and purpose of HTML, Role of HTML in web development, Basic syntax and structure
Setting Up Your Development Environment: Text editors (VS Code, Sublime, Notepad++), Browser for testing HTML (Google Chrome, Firefox)
First HTML Page: Creating and saving a basic HTML document, Introduction to <!DOCTYPE html> and basic document structure, Tags: <html>, <head>, <body>
Head Section: The <meta> tag, The <title> tag, The importance of the <head> section
Body Section: Basic text elements: <h1> to <h6>, <p>, <b>, <i>, Paragraphs, line breaks, and horizontal rules, Lists: Ordered (<ol>) and unordered (<ul>) lists, list items (<li>)
Hyperlinks and Images: Creating hyperlinks with <a> tag, Attributes: href, target, title, Adding images with <img> tag, Attributes: src, alt, width, height
Forms and Input Elements: Creating forms: <form> tag, Form elements: <input>, <textarea>, <select>, <button>, Basic form attributes: action, method, name, id
Tables: Table tags: <table>, <tr>, <th>, <td>, Styling tables: border, cellspacing, cellpadding
Multimedia Elements: Embedding audio: <audio>, Embedding video: <video>, Using <source> tag for multiple file formats
Iframes and Embeds: Creating iframes with <iframe>

MODULE II: HTML5 AND Advanced HTML5 Features

HTML5 Overview: Introduction to HTML5 features, Key differences between HTML5 and previous versions, Benefits of HTML5 for modern web development
Semantic HTML5 Tags: Introduction to semantic tags: <header>, <footer>, <nav>, <article>, <section>, <aside>, etc.

Audio and Video Elements in HTML5: Using <audio> and <video> tags, Supporting multiple file formats and fallback methods
Forms in HTML5: New input types: email, date, number, range, url, etc., New attributes: required, placeholder, autofocus, pattern, Creating a form with new HTML5 inputs
Local Storage and Session Storage: Introduction to local storage and session storage, Storing and retrieving data with JavaScript, Use cases for local storage in web applications
HTML5 Canvas: Introduction to <canvas> element, Drawing basic shapes and graphics with JavaScript, Animation basics using canvas

MODULE-III: Mastering CSS3

What is CSS: Definition and purpose of CSS, The role of CSS in web development, How CSS works with HTML to style web pages
CSS Syntax and Structure: CSS rule sets: selectors, properties, and values, Basic syntax: selector { property: value; }, Inline, internal, and external CSS
How to Link CSS to HTML: Inline CSS using the style attribute, Internal CSS within the <style> tag, External CSS with the <link> tag
CSS Selectors: Element selectors (e.g., h1, p), Class selectors (e.g., .class-name), ID selectors (e.g., #id-name), Universal selector (*), Descendant, child, and sibling selectors, Attribute selectors (e.g., [type="text"])
CSS Properties: **Text styling:** color, font-family, font-size, line-height, font-weight, text-align, **Box model:** width, height, margin, padding, border, box-sizing, **Backgrounds:** background-color, background-image, background-position, background-repeat
Understanding the Box Model: Content, padding, border, and margin, Box-sizing property (content-box, border-box), Visualizing the box model using browser developer tools
Layout Techniques: Static vs. relative vs. absolute positioning, Floating elements and clearing floats, Centering elements using margin auto, Fixed positioning and sticky positioning
CSS3 Transitions: What are CSS transitions?, Transition properties: transition-property, transition-duration, transition-timing-function, transition-delay, Example: Hover effect with transitions
CSS3 Animations: Introduction to keyframes, Creating animations with @keyframes, Animation properties: animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, Practical examples of animations
CSS3 Transforms: transform property: rotate, scale, translate, skew, 2D vs. 3D transforms, Transforming elements on hover (interactive effects)

MODULE-IV: CSS3 Responsive Design

Introduction to Responsive Web Design (RWD): What is responsive design?, Importance of mobile-first design, Viewport meta tag and its importance in mobile optimization
Media Queries: Syntax of media queries, Targeting different devices and screen sizes, Example: Mobile-friendly layout using media queries
Fluid Layouts and Flexible Boxes: Using percentages for fluid widths, Introduction to Flexbox: Basic layout with display: flex, Flexbox properties: justify-content, align-items, flex-wrap, flex-grow, flex-shrink, flex-basis
CSS Grid Layout: Introduction to the CSS Grid system, Defining grid containers with display: grid, Creating rows and columns with grid-template-rows, grid-template-columns, Aligning grid items using justify-items, align-items, and place-items, Example: Building a responsive grid layout
CSS Flexbox and Grid Combined: Combining Flexbox and Grid for complex layouts, Practical use case: Building a multi-column layout with both Flexbox and Grid
Custom Properties (CSS Variables): Introduction to CSS custom properties (variables), Defining and using variables: --primary-color, etc., Benefits of using CSS variables in themes and design consistency
Hover and Focus Effects: Styling links and buttons on hover and focus, Changing background colors, borders, and text styles, Example: Button hover effects with transitions

CSS Shadows and Glows: Box shadows: box-shadow property, Text shadows: text-shadow property, Using multiple shadows in one property, Creating glowing effects with shadows
Gradients and Patterns: Linear gradients: background: linear-gradient (), Radial gradients: background: radial-gradient(), Repeating gradients and patterns, Example: Background gradient animations
Styling Form Elements: Basic form styles: input fields, buttons, and labels, Input types and custom styles for text, password, email, number, etc., Placeholder and focus effects
Customizing Form Controls: Styling checkboxes, radio buttons, and select dropdowns, Custom form controls with appearance property, Example: Custom form controls with CSS
CSS3 Validation Styles: Styling invalid and valid form inputs, Using the: valid, invalid, required, and: focus pseudo-classes

MODULE-V: Mastering JavaScript

What is JavaScript: Definition and role of JavaScript in web development, JavaScript's relationship with HTML and CSS, Running JavaScript: Inline, Internal, and External JavaScript
JavaScript Syntax Basics: Variables: let, const, and var, Basic data types: string, number, boolean, null, undefined, Simple operators: +, -, *, /, %, ++, --, Comments: single-line and multi-line comments
Conditional Statements: if, else if, else, switch statement
Loops: for loop, while loop, do...while loop, for...of and for...in loops, Loop control: break, continue
Functions: Function declaration and invocation, Parameters and arguments, Return values, Function expressions and arrow functions (() => {}), Scope: Local and Global, Closures and higher-order functions
Arrays: Declaring arrays: let arr = [], Array methods: push(), pop(), shift(), unshift(), slice(), splice(), map(), filter(), reduce(), Accessing and iterating through arrays: for, forEach()
Objects: Declaring objects: let obj = { key: value }, Accessing and modifying object properties: dot notation and bracket notation, Iterating through objects: for...in loop, Object.keys(), Object.values()
Introduction to the DOM: What is the DOM? (Document Object Model), Accessing elements by ID, class, tag, and query selectors, Modifying content and attributes: innerHTML, textContent, setAttribute()
Event Handling: Adding event listeners: addEventListener(), Handling different events: click, mouseover, keydown, etc., Event propagation: event bubbling and event capturing
ES6+ Features: Let and const vs. var, Arrow functions and the this keyword, Template literals (String interpolation): `Hello, ${name}!`, Default parameters in functions, Modules: import and export
Asynchronous JavaScript: Callbacks, Promises: then(), catch(), finally(), Async/Await: Simplifying asynchronous code
Error Handling in JavaScript: try...catch block, Throwing errors with throw, Custom error messages, Handling asynchronous errors
Debugging JavaScript: Using browser dev tools (Console, Sources, Breakpoints), Common debugging techniques, Debugging asynchronous code

MODULE-VI: Mastering Bootstrap

What is Bootstrap: Overview of Bootstrap and its importance, Advantages of using Bootstrap for responsive web design, Installing Bootstrap (via CDN and local setup)
Setting Up Your Development Environment: Text editors: VS Code, Sublime Text, etc., Browsers and developer tools for testing, Linking Bootstrap to your project: CDN vs. downloading
Bootstrap Grid System: Understanding the 12-column grid layout, Defining rows and columns with .row and .col-*, Creating responsive layouts using grid breakpoints, Nesting grid columns for advanced layouts
Typography: Bootstrap's default typography styles (headings, paragraphs, lists), Font styles and text alignment: .text-center, .text-left, .text-right, Typography utilities: .font-weight-bold, .text-uppercase, etc.

Buttons: Button classes: .btn, .btn-primary, .btn-success, .btn-danger, Button sizes and block buttons, Button groups and toolbar buttons, Button states: active, disabled, focus, and hover
Images: Responsive images with .img-fluid, Rounded images, circles, and thumbnails with .rounded, .rounded-circle, .img-thumbnail, Image alignment and utilities
Icons with Bootstrap: Using Bootstrap Icons or integrating Font Awesome, Applying icons to buttons and links, Icon sizes and alignment
Containers: .container vs. .container-fluid, Fixed-width vs. full-width containers, Responsive containers for different screen sizes
Bootstrap Grid System in Detail: Creating multi-column layouts: .col-md-6, .col-lg-4, etc., Offsetting and ordering columns, Grid nesting and offsets for complex layouts
Spacing Utilities: Margins and paddings: .m-3, .mt-5, .px-2, Responsive spacing classes, Controlling spacing between elements with margin/padding utilities
Navigation Bar (Navbar): Creating responsive navigation bars using .navbar, Navbar components: brand, links, forms, and dropdowns, Mobile-friendly navbar using the hamburger menu, Customizing navbar background, colors, and alignments
Cards: Using .card for creating card components, Card body, header, footer, and image, Card groups and card decks for multi-card layouts, Styling cards with custom classes
Alerts: Using .alert for displaying messages, Customizing alert styles: success, warning, danger, info, Dismissing alerts with JavaScript and .alert-dismissible
Modals: Creating modal dialogs with .modal, Modal header, body, footer, and close button, Controlling modal visibility using JavaScript, Using modal for forms, notifications, and more
Forms: Creating forms with .form-control, Input groups for adding icons or buttons to form fields, Custom checkboxes, radio buttons, and selects, Form validation using Bootstrap classes and custom styles
Display and Positioning: Display utilities: .d-block, .d-inline, .d-none, etc., Visibility utilities: .visible, .invisible, Positioning utilities: .position-relative, .position-absolute, .position-fixed
Flexbox Utilities: Flexbox basics: .d-flex, .justify-content-*, .align-items-*, .flex-row, .flex-column, Flexbox utilities for centering and alignment, Flexbox ordering and wrapping
Colors and Backgrounds: Text and background color utilities, Color classes: .text-primary, .bg-success, etc., Background utilities for gradients and images
Sizing and Overflow Utilities: Width and height utilities: .w-50, .h-100, Controlling overflow: .overflow-auto, .overflow-hidden
Borders and Shadows: Border utilities: .border, .border-top, .rounded, .border-light, Box-shadow utilities: .shadow, .shadow-lg
Mobile-First Design Philosophy: Why Bootstrap is mobile-first and how it helps responsive design, Understanding responsive breakpoints (xs, sm, md, lg, xl, xxl)
Creating Responsive Layouts: Building a mobile-first, responsive webpage, Making images, tables, and forms responsive, Handling mobile navigation with collapsible menus
Carousel: Creating image carousels with .carousel, Carousel controls: next/previous buttons and indicators, Customizing carousel items and controls

INTRODUCTION TO NODE JS
INTRODUCTION TO EXPRESS JS

Project 1 – Responsive Landing Page: Build a simple, responsive landing page with navigation, hero section, and call-to-action button
Project 2 – Blog Layout: Create a responsive blog layout with cards, grid system, and a sidebar navigation
Project 3 – E-commerce Product Page: Design a product page with cards, product details, image gallery, and a modal for product options
Project 4 – Dashboard Layout: Create a responsive admin dashboard with navigation, grid-based layout, cards, and data tables

Text Book:
Introduction to frontend technology HTML, CSS and JavaScript Web Publishing language By Laura Lemay, Rafe Colburn, Jennifer kyrnin 1st Edition,2022

Reference Books:
Theory, practical and critical problem solving in frontend technology (HTML5, CSS3, JavaScript and Adv. JavaScript): Dr. T. Vasudev, Dr. Chandrajit M & Prof. Arvind G.

Course Outcomes:

After completing the course, students will be able to:

1. Students will be able to identify the different computational problems and their associated complexity.
2. Students will be able to differentiate and give examples for the different
3. types of automata like finite automata, push down automata, linear bounded automata and Turing machine.
4. To apply the techniques of designing grammars and recognizers for several programming languages.
5. Students will be able to correlate the different types of automata to real world applications.

BCAICC407: Java Programming						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite:

1. C programming
2. Object Oriented Programming using C++

Course Objectives:

1. To introduce the fundamental concepts of object-oriented programming (OOP) and how they are implemented in Java.
2. To develop the ability to write, compile, and debug Java programs using classes, objects, inheritance, polymorphism, and interfaces.
3. To familiarize students with Java's standard libraries, exception handling, file I/O, and multithreading capabilities.
4. To enable students to build GUI-based applications and real-world projects using Java frameworks and tools.

Detailed Syllabus

Unit I: Introduction

History and Overview of Java, Object Oriented Programming, Control statements- if and for loop. Using Blocks of codes, Lexical issues - White space, identifiers, Literals, comments, separators, Java Key words, Data types - Integers, Floating point, characters, Boolean, a closer look at Literals, Variables, Type conversion and casting. Automatic type promotion in Expressions Arrays. Operators - Arithmetic operators, Bit wise operators, Relational Operators, Boolean Logical operators, Assignment Operator, Operator Precedence. Control Statements – Selection Statements - if, Switch, Iteration Statements - While, Do-while, for Nested loops, Jump statements.

Unit II: Classes

Class Fundamentals, Declaring objects, Assigning object reference variables. Methods - constructors, “this” keyword, finalize ( ) method A stack class, Over loading methods. Using objects as parameters, Argument passing, Returning objects. Recursion, Access control, Introducing final, understanding static. Introducing Nested and Inner classes. Using command line arguments. Inheritance – Basics, Using super, method overriding, and Dynamic method Dispatch, Using abstract classes and final with Inheritance.

Unit III: Packages

Definition. Access protection importing packages. Interfaces: Definition and implementation. Exception Handling – Fundamentals, types, Using try and catch and Multiple catch clauses, Nested try Statements, throw, throws, finally. Java’s built-in exception, using Exceptions.

Unit IV: Multithreaded Programming:
Java thread model – main thread, creating single and multiple thread. Is alive ( ) and join ( ). Thread – Priorities, Synchronization, Inter thread communication, suspending, resuming and stopping threads, using multi-threading. I / O basics – Reading control input, writing control output, Reading and Writing files. Applet Fundamentals – AWT package, AWT Event handling concepts, the transient and volatile modifiers. Using instance of using assert.

Unit V: Java Networking and GUI Application Development:
Java Networking: Concept of client and Server, Introduction of TCP, Concept of Socket, Importance of Socket, Socket programming, communication between client and server.
GUI Application Development: Introduction to AWT, AWT controls Java Applet, Layout Managers, Menus, Images, Graphics, Event Handling, Swing, Containers, Panes, Frames, Dialogue boxes, working with image controls.

Unit VI: JAVA Database Connectivity (JDBC)
Database connectivity – JDBC architecture and Drivers. JDBC API - loading a driver, connecting to a database, creating and executing JDBC statements, handling SQL exceptions. Accessing result sets: types and methods. An example - JDBC application to query a database. .

Text and Reference Books:

1. The Complete Reference Internet, Margaret Levine Young, TMH, 1999.
2. The Complete Reference JAVA 2, Naughton Schildt, TMH, 5th Edition.
3. Programming in JAVA, E. Balagurusamy E, TMH, 3rd Edition, 2006.
4. Java Black book, Steven Helzner, Dreamtech , 2002

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts of object-oriented programming using Java.
CO2: Apply Java programming constructs such as classes, objects, inheritance, and polymorphism to solve real-world problems.
CO3: Develop Java applications using packages, interfaces, exception handling, and multithreading.
CO4: Implement GUI-based applications using AWT and Swing components.
CO5: Use Java for database connectivity (JDBC) and develop simple client-server applications.

BCAICC408: Logical Reasoning and Thinking						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
1	1	0	35	15	50	1

Teaching Scheme	**Examination Scheme**
Credits: 2	Mid Term Exam: 6 Marks
	Teachers Assessment: 3 Marks
	Attendance: 6 Marks
	End Semester Exam: 35 Marks

Prerequisite: Basic Analytical and Critical Thinking Skills, Understanding of Fundamental Mathematical and Verbal Concepts

Course Objectives:

1. To develop students' ability to analyze and evaluate arguments logically and critically.
2. To enhance problem-solving skills through various logical reasoning techniques.
3. To familiarize students with different types of reasoning such as deductive, inductive, and analytical reasoning.
4. To prepare students for aptitude tests, competitive exams, and real-world decision-making scenarios.

Detailed Syllabus

Unit I: Verbal ability

Synonyms, Antonyms and One word substitutes, Speed, Time and Distance, Time and Work, Linear Equations, Progressions (Sequences & Series), Permutation and Combination, Probability, Functions, Set Theory, Number Systems, LCM and HCF, Percentages, Collection and Scrutiny of data: Primary data, questionnaire and schedule; secondary data, their major sources including some government publications.

Unit II: Logical Reasoning - I

Number and Letter Series, Calendars, Clocks, Cubes, Venn Diagrams, Binary Logic, Seating Arrangement, Logical Sequence, Logical Matching, Logical Connectives, Syllogism, Blood Relations; concept of a statistical population and sample from a population; qualitative and quantitative data

Unit III: Presentation of Data

Construction of tables with one or more factors of classification; Diagrammatic and Graphical representation of non-frequency data; Frequency distribution, cumulative frequency distribution and their graphical representation - histogram, Column Graphs, Bar Graphs, Line Charts, Pie Chart, Data Interpretation – Introduction and approaches

Text Books:

1. Richard I Levin, David S. Rubin: Statistics for Management, Pearson Prentice Hall Education Inc. Ltd, NewDelhi, 5th Ed. 2007
2. Bajpai, N. Business Statistics, Pearson, 2010.

Reference Books:

1. Sharma J.K., Business Statistics, Pearson Education India, 2010.
2. Anderson; David R, Dennis J. Sweeney and Thomas A. Williams, Quantitative Methods for Business, Prentice-Hall, West Publishing Company, 1996.
3. CAT Complete course, UPKAR publications.

Course Outcome:

After completing the course, students will be able to:

CO1: Apply logical thinking to solve problems using deductive and inductive reasoning methods.
CO2: Analyze and evaluate complex arguments for validity and soundness.
CO3: Identify logical fallacies and improve critical thinking abilities in academic and professional contexts.
CO4: Demonstrate proficiency in solving puzzles, syllogisms, coding-decoding, and analytical reasoning problems.
CO5: Prepare effectively for competitive exams and aptitude-based assessments through structured logical reasoning techniques.

BCA501: MERN Full Stack Development- Backend						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	4	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Pre-requisites: DBMS, Manipulate files and installation of software.

Course Objectives:

1. Understand JavaScript fundamentals, including variables, data types, functions, loops, operators, and the Event Loop, to develop basic programming solutions.
2. Apply Advanced JavaScript concepts like arrays, objects, ES6 features, DOM manipulation, asynchronous programming, and API handling using fetch and Axios.
3. Understand the core concepts of Node.js, including its asynchronous, event-driven nature. Learn how to use Nod e.js to build high-performance, non-blocking backend services.
4. Understand the fundamentals of MongoDB, a NoSQL database, and its advantages in modern applications. Learn how to perform basic operations like create, read, update, and delete (CRUD) in MongoDB.
5. Learn Mongoose, an ODM (Object Data Modeling) library that simplifies interaction with MongoDB. Understand how to define schemas, models, and validation for MongoDB documents.

Detailed Syllabus:

Unit-I
Introduction to JavaScript, Features, Scopes, Variables- Var, Let and Const, Data Type, Conditional Statements, Operators, Loops, Comments, Function, Anonymous Function, Map, Filter, Reduce, Event Loop.

Unit-II
Array, Object, Array and Object destructuring, ES6, Dom Manipulation, Error Handling, Asynchronous JavaScript- Promises, Callbacks, Async Await. Introduction to API – Fetch and AXIOS, JSON – Parse, Stringify

Unit-III
Introduction to React, Introduction to NodeJS, what is node.js, Introduction to NPM, Node Process Model, Setup node.js, Debugging node.js app, Modules in NodeJS, understanding exports and require, Creating modules

Unit-IV
Express JS, Introduction to Express JS, Installation of Express JS, Creating server using express JS, Web Browser Building, Express Router, Express Listen Methode, Request matching, Route parameter, Handler function, Request object and Response object, get, post, put, delete Methods, Packages and middleware, Body-parser, cors and credential,

Unit-V

Introduction to MongoDB, Installing MongoDB, The current SQL/NoSQL landscape, Document oriented vs. other types of storage, MongoDB databases, MongoDB Collections, MongoDB Documents, CRUD Operations in MongoDB, MongoDB Methods- insert(), update(), save(), find(), delete(), Working with equality, Query operators, Building complex queries, Updating documents, Deleting documents

Unit-VI

Introduction to Mongoose, installing Mongoose, Connecting to MongoDB from Mongoose, Core concepts of Mongoose, understanding Mongoose schemas and datatypes, Working with Models, Using modifiers in schema, Using virtual fields, and Optimizing query performance by enabling indexes.

Text Book:

1. Introduction to Backend Development Technology “Node.js” Design Patterns" by Mario Casciaro
2. "Express.js Guide: The Comprehensive Book on Express.js" by Azat Mardan
3. "Learning MongoDB" by Jason O'Brien, "Mongoose for Application Development" by Simon Holmes

Reference Books:

1. Theory, practical and critical problem solving in backend technology (Node JS, Express JS, Mongoose and MongoDB): by Mr. David Herron, Mr. Douglas Crockford, Mr. Kristina Chodorow

.

Course Outcomes:

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

1. Students will be able to define the programming principles behind theoretical computer science.
2. Students will be able to identify the different computational problems and their associated complexity.
3. Students will understand how to connect client with server, creating a server and HTTP methods.
4. Students will be able to differentiate and give examples for the different types of backend model , Handle file uploads, interactions with cloud services, and external databases.
5. Learn how to use MongoDB to perform advanced data querying, validation, and population (relationships between documents).
6. Integrate the MERN stack (MongoDB, Express.js, React.js, Node.js) and apply deployment strategies to build and host full-stack applications effectively.

BCAI502: Block Chain Technology						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
3	1	0	70	30	4	100

Teaching Scheme	**Examination Scheme**
Credits: 4	Mid Term Exam: 12 Marks
	Teachers Assessment: 6 Marks
	Attendance: 12 Marks
	End Semester Exam: 70 Marks

Prerequisite: Advanced Computer Networks, Cryptography and Network Security.

Course Objectives:

1. To introduce the fundamental concepts and architecture of blockchain technology, including blocks, cryptographic hash functions, and distributed ledgers.
2. To explain the working of consensus mechanisms such as Proof of Work, Proof of Stake, and their role in maintaining integrity in decentralized systems.
3. To familiarize students with blockchain platforms like Bitcoin, Ethereum, and smart contract development using tools such as Solidity.
4. To enable students to analyze and implement real-world blockchain applications in finance, supply chain, healthcare, and other industries.

Detailed Syllabus:

Unit-1

Introduction to Blockchain: History of centralized services, trusted third party for transactions, understand the difference between centralized, decentralized and distributed peer to peer networks, why Block chain?, Types of Blockchain.

History of Bitcoins: How and when Blockchain and Bitcoin started. Milestone on the development of bitcoin, Problem area of Bitcoin, relation to Bitcoin, requirement of block chain in a business environment, sharing economy, requirements deep dive, Internet of value.

Unit-2

Consensus: Mechanism, Types of Consensus Mechanism, Consensus in Blockchain. Decentralization: Disintermediation and Contest Driven Decentralization, Routes to Decentralization, Full Ecosystem Decentralization, Smart Contracts, Decentralized Organizations, Platforms for Decentralization.

Unit-3

Blockchain Applications and USE case: Business drivers of blockchain, Digital currency and finance (including ICOs and alternative funding), Identity, Supply Chain, Healthcare, Ownership and property rights Governance and compliance.

Unit-4

Blockchain Challenges and Constraints: Blockchain risks, Technological challenges, standards Scalability issues, Security and privacy, Legal and regulatory problems, Social and cultural constraints.

Unit-5

Ethereum: Ethereum network, EVM, Transaction fee, Mist, Ether, gas, Solidity - Smart contracts, Truffle, Web3, Design and issue Cryptocurrency, Mining, DApps, DAO.

Unit-6

Introduction to Hyperledger Fabric: What is Hyperledger, Why Hyperledger, Where can Hyperledger be used, Hyperledger Architecture, Membership, Blockchain, Transaction, Chaincode, Hyperledger Fabric, Features of Hyperledger, prerequisiteof Fabric installation

Suggested Readings:

1. A. Narayanan, J. Bonneau, E. Felten, A. Miller & S. Gold feder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
2. B. Singhal & G. Dhameja Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions, Apress 2018.
3. D. Mohanty, Blockchain - From Concept to Execution, BPB Publications, 2018.
4. Imran Bashir, Mastering Blockchain, 2nd Edition, Packt Publishing, 2018.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental principles, architecture, and components of blockchain systems.
CO2: Analyze different consensus mechanisms (e.g., Proof of Work, Proof of Stake) and their role in securing blockchain networks.
CO3: Develop and deploy smart contracts using platforms like Ethereum and programming languages such as Solidity.
CO4: Evaluate the applicability of blockchain in various sectors such as finance, supply chain, healthcare, and digital identity.
CO5: Demonstrate the ability to design simple decentralized applications (DApps) and understand key challenges in blockchain scalability and security.

BCAI513: Server Side Scripting Language						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic understanding of HTML, CSS, and Client-Side Scripting (e.g., JavaScript), Fundamental knowledge of Programming Concepts and Database Management Systems (DBMS).

Course Objectives:

1. To introduce the fundamentals of server-side scripting and web server communication.
2. To enable students to develop dynamic and interactive web applications using server-side technologies.
3. To provide knowledge of handling form data, session management, and database connectivity.
4. To develop the ability to integrate server-side scripts with client-side technologies for full-stack web development.

Detailed Syllabus

Unit 1: Introduction to Server-Side Scripting
Basics of Web Architecture: Client vs Server, Static vs Dynamic Web Pages, Introduction to Server-Side Scripting Languages (PHP, Node.js, Python, etc.), HTTP Request and Response Lifecycle, Role of Web Servers (Apache, Nginx), Setting up Server-Side Environment (XAMPP, WAMP, LAMP)

Unit 2: Core PHP / Server Language Syntax & Functions
Syntax, Variables, Data Types, Constants, Operators and Expressions, Control Structures: if-else, switch, loops (for, while, foreach)
Functions: User-defined and Built-in, Arrays: Indexed, Associative, Multidimensional, Include and Require Statements, Error Handling and Debugging

Unit 3: Form Handling and User Input
HTML Forms and POST/GET Methods, Form Validation (Server-Side), Secure Handling of User Inputs, File Uploading and Processing, Session Management and Cookies, Redirects and Headers

Unit 4: Database Connectivity and Advanced Server-Side Features
Introduction to MySQL / PostgreSQL, CRUD Operations (Create, Read, Update, Delete) using SQL, Connecting Server Scripts to Database (MySQLi / PDO in PHP or equivalent), Prepared Statements and SQL Injection Prevention, Displaying Database Results in Web Pages, Error Handling in Database Interactions
Advanced Server-Side Features: Working with JSON and XML Data, File Handling: Read, Write, Append, Authentication and Authorization, Email Sending Functionality, Introduction to REST APIs and Web Services, Uploading Files to Server and Handling MIME Types

Unit 5: Security and Deployment in Server-Side Scripting
Common Web Vulnerabilities: SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Session Hijacking

Securing Server-Side Code: Input Validation and Sanitization, Secure Authentication Practices, Password Hashing (e.g., bcrypt, SHA), HTTPS and SSL Certificates, Error Reporting and Logging Mechanisms
Deployment Best Practices: Hosting Options for Server-Side Applications ,Environment Variables and Configuration Files, Version Control and Continuous Deployment (Git basics)
Performance Optimization Techniques: Caching (server-side and browser-side), Load Balancing Concepts, Compression (GZIP)

Text and Reference Book

1. Luke Welling & Laura Thomson, PHP and MySQL Web Development, Pearson Education, 5th Edition.
2. Robin Nixon, Learning PHP, MySQL & JavaScript, O'Reilly Media, 5th Edition.
3. Kevin Yank, Build Your Own Database Driven Website Using PHP & MySQL, SitePoint.
4. Steven Holzner, PHP: The Complete Reference, McGraw Hill Education.

Course Outcomes:

After completing the course, students will be able to

CO1: Understand the fundamentals of server-side scripting and its role in dynamic web development.
CO2: Develop interactive web applications using PHP and integrate them with backend databases.
CO3: Apply form handling, session tracking, and error handling techniques in server-side scripts.
CO4: Utilize MySQL or similar RDBMS to perform CRUD operations securely from PHP scripts.
CO5: Design, test, and deploy full-featured, database-driven websites using server-side technologies.

BCAI514: Android Application Development						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: - Basics of Java language and PL/SQL

Course Objectives:

1. To introduce students to the Android platform, architecture, and development environment using Android Studio.
2. To develop the ability to design and build interactive and user-friendly mobile applications using Java/Kotlin and XML.
3. To enable students to utilize Android components such as Activities, Intents, Services, Broadcast Receivers, and Content Providers.
4. To provide hands-on experience in integrating mobile apps with databases (e.g., SQLite) and web services (e.g., REST APIs).

Detailed Syllabus

UNIT I

JAVA Concepts (10 hrs): Platform Independency, OOPs Concepts, Inheritance in detail, Exception handling, Packages & interfaces, JVM & .jar file extension, Multi threading (Thread class & Runnable Interface).SQL: DML & DDL Queries in brief.

UNIT II

Introduction to Android: Introduction of Android, Setting up development environment, Installing the SDK, Creating Android Emulator, Android development Tool. Fundamentals: Basic Building blocks - Activities, Services, Broadcast Receivers & Content provider, UI Components - Views & notifications, Components for communication -Intents & Intent Filters, Android API levels (versions & version names)

UNIT III

Application Structure: AndroidManifest.xml, uses-permission & uses-sdk, Resources & R.java, Assets, Layouts & Draw-able Resources, Activities and Activity lifecycle, First sample Application.

UNIT IV

Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS. Second App: (switching between activities), Develop an app for demonstrating the communication between Intents.

UNIT V

Basic UI design: Form widgets, Text Fields, Layouts, [dip, dp, sip, sp] versus px, Examples
Preferences: Shared Preferences, Preferences from xml, Examples.
Menu: Option menu, Context menu, Sub menu, Menu from xml, Menu via code, Examples
UI design: Time and Date, Images and media, Composite, Alert Dialogs & Toast, Popup, Examples

Text and Reference Books

1. Android Application Development (With Kitkat Support), Black Book, by Kogent Learning Solutions Inc. by Pradeep Kothari
2. Android Application Development Cookbook: 93 Recipes for Building Winning Apps (WROX),by Wei-Meng Lee
3. Professional Android 4 Application Development, by Reto Meier
4. Beginning Android 4 Application Development, Wei-Meng Lee
5. Android Application Development, by Lombardo John and Blake Meike

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamentals of Android architecture, lifecycle, and development environment.
CO2: Design and develop user interfaces using Android UI components and layouts.
CO3: Implement core Android components such as Activities, Services, Broadcast Receivers, and Content Providers.
CO4: Develop Android applications that interact with internal (SQLite) and external databases or APIs.
CO5: Build and deploy functional Android apps with features like data storage, multimedia, and location services.

BCAI515 (Elective-IV): Introduction to Web Services						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	0	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Web Technologies, Fundamentals of Programming

Course Objectives:

1. To understand the fundamental concepts and architecture of web services, including SOAP, REST, WSDL, and UDDI.
2. To enable students to design, develop, and deploy web services using standard protocols and frameworks.
3. To explore interoperability between distributed applications through web services using platform-independent communication.
4. To apply web service technologies in real-world applications, ensuring secure, scalable, and efficient service-oriented solutions.

Detailed Syllabus

UNIT I:
Evolution and Emergence of Web Services: Evolution of distributed computing, Core distributed computing technologies, client/server, CORBA, JAVA RMI, Micro Soft DCOM, MOM, Challenges in Distributed Computing, role of J2EE and XML in distributed computing, emergence of Web Services and Service Oriented Architecture (SOA). Introduction to Web Services: The definition of web services, basic operational model of web services, tools and technologies enabling web services, benefits and challenges of using web services. Web Services Architecture: Web services Architecture and its characteristics, core building blocks of web services, standards and technologies available for implementing web services, web services communication, basic steps of implementing web services.

UNIT II:
Fundamentals of SOAP: SOAP Message Structure, SOAP Encoding, Encoding of different data types, SOAP message exchange models, SOAP communication and messaging, Java and Axis, Limitations SOAP.

UNIT III:
Describing Web Services: WSDL, WSDL in the world of Web Services, Web Services life cycle, anatomy of WSDL definition document, WSDL bindings, WSDL Tools, limitations of WSDL.

UNIT IV:
Discovering Web Services: Service discovery, role of service discovery in a SOA, service discovery mechanisms, UDDI: UDDI Registries, uses of UDDI Registry, Programming with UDDI, UDDI data structures, Publishing API, Publishing information to a UDDI Registry, searching information in a UDDI Registry, limitations of UDDI.

UNIT V:

Web Services Interoperability: Means of ensuring Interoperability, Overview of .NET, Creating a .NET Client for an Axis Web Services, Creating Java Client for a web service, Challenges in Web Services Interoperability. Services Security: XML security frame work, Goals of cryptography, Digital Signature, Digital Certificate, XML encryption.

Text Books:

1. Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley India, rp 2008.
2. Developing Enterprise Web Services, S. Chatterjee, J. Webber, Pearson Education, 2008.
3. XML, Web Services, and the Data Revolution, F.P.Coyle, Pearson Education.

Reference Books:

1. Building Web Services with Java, Second Edition, S. Graham and others, Pearson Edn., 2008.
2. Java web services, D.A. Chappell and T.Jewell, O'Reilly,SPD.
3. Java Web Services Architecture, McGovern, Sameer Tyagi etal.., Elsevier.
4. Web Services, G. Alonso, F. Casati and others, Springer, 2005.

Course Outcomes:

After completing the course, students will be able to:

CO1: Demonstrate understanding of web service protocols and standards such as SOAP, REST, WSDL, and UDDI.
CO2: Design and develop interoperable web services using appropriate technologies and tools.
CO3: Implement RESTful and SOAP-based web services in a secure and scalable manner.
CO4: Analyze and apply XML and JSON data formats in the context of web services communication.
CO5: Evaluate the performance and security aspects of web services in distributed computing environments.

BCAI516 (Elective-IV): Cyber Law and Information Security						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Computer Networks and Internet Technologies, Fundamentals of Information Technology

Course Objectives:

1. To provide an understanding of cyber laws and legal frameworks governing cyberspace, data protection, and digital rights.
2. To familiarize students with key information security concepts, including confidentiality, integrity, and availability of data.
3. To analyze various types of cybercrimes and legal remedies, including policies and penalties under IT Act and international laws.
4. To equip students with knowledge of security standards, tools, and ethical practices in securing digital systems and networks.

Detailed Syllabus

Unit I: Introduction to Cyber Law

Introduction to Cyber Law: Overview of Cyberspace and Cyber Law, Need for Cyber Law in the Digital Age, History and Evolution of Cyber Law, Legal Concepts Related to Information Technology, Jurisdiction in Cyberspace, Overview of IT Act, 2000 and Amendments

Unit II: Information Security Fundamentals

Information Security Concepts: CIA Triad, Types of Threats and Attacks: Malware, Phishing, DoS, etc., Security Goals and Principles, Risk Management and Security Policies, Introduction to Cryptography and Authentication, Basics of Network Security and Firewalls

Legal Framework and IT Act Provisions: Key Provisions of the IT Act, 2000, Digital Signature and Electronic Records, Cyber Offenses and Penalties, Cyber Regulations Appellate Tribunal, Role of Certifying Authorities, Case Studies under IT Act

Unit III: Cyber Crimes and Legal Issues

Classification of Cyber Crimes: Hacking, Cyber Stalking, Cyber Terrorism, Intellectual Property Issues: Copyright, Trademark, Domain Names, Data Protection and Privacy Laws, Cyber Forensics and Evidence Collection, International Cyber Laws and Treaties, Case Studies on Cyber Crime Investigations

Unit IV: Information Security Management and Standards

Security Policy and Governance, Risk Assessment and Incident Response, ISO 27001, NIST Framework, and Other Standards, Secure Software Development Practices, Role of Ethics and Professional Responsibility, Business Continuity and Disaster Recovery Planning

Unit V: Emerging Issues and Future Trends

Cloud Computing Security and Legal Challenges, Social Media, Mobile Security, and Privacy Concerns, AI and IoT Security Implications, Cybersecurity Laws in Other Countries, Cyber Law Compliance for Organizations, Career Opportunities in Cyber Law and Information Security

Text and Reference Book

1. R.K. Suri & T.N. Chhabra – Cyber Crime and Information Technology Laws, Bharat Law House
2. Pankaj Duggal – Cyber Law and Information Security, Katson Publishing House
3. Justice Yatindra Singh – Cyber Laws, Universal Law Publishing Co.
4. V.K. Jain - Cyber Crime and Information Security, Dreamtech Press

Course Outcomes:

After completing the course, students will be able to

CO1: Understand the fundamental concepts of cyber law and the legal framework governing cyberspace.
CO2: Analyze and interpret key provisions of the Information Technology Act, 2000 and its amendments.
CO3: Identify various forms of cybercrimes and apply appropriate legal and technical measures to address them.
CO4: Demonstrate knowledge of information security principles, tools, and risk management practices.
CO5: Evaluate the ethical, legal, and social implications of emerging cybersecurity issues and global cyber laws.

BCAI517 (Elective-IV): Embedded Systems						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
2	1	2	50	25	2	75

Teaching Scheme	**Examination Scheme**
Credits: 3	Mid Term Exam: 10 Marks
	Teachers Assessment: 5 Marks
	Attendance: 10 Marks
	End Semester Exam: 50 Marks

Prerequisite: Basic Knowledge of Digital Electronics, Fundamentals of Programming (Preferably in C/C++)

Course Objectives:

1. To introduce the basic concepts and architecture of embedded systems, including microcontrollers and their applications.
2. To develop an understanding of embedded hardware and software components and their interaction in real-time environments.
3. To equip students with skills to design, program, and debug embedded systems using appropriate tools and platforms.
4. To analyze and apply embedded system concepts in designing solutions for automation, IoT, and industry-specific applications.

Detailed Syllabus

Unit 1:
Introduction to Embedded System: History & need of Embedded System, Basic components of Embedded System, Programming Language Classification of Embedded System, Advantage & Disadvantage **Microprocessor & Microcontroller Classification:** Difference between Microprocessor & Microcontroller, Classification based on architecture, Memory Classification

Unit 2:
Introduction of Embedded C: Introduction to Embedded C, Difference between C & Embedded C, Programming style, Basic structure of C Program **Constants, Variables & Data Types:** Keywords & Identifiers, Data type & its memory representation, Arrays and strings **Operators:** Types of Operators, Bitwise Operators explained

Unit 3:
Control Structures & Loops: Decision making with if statement, If…. else statement, Switch statement, and GOTO statement, The While and Do – While statements, for statement **Functions:** Why Functions, Types of Functions, A Multi-functional program, Return values & their types

Unit 4:
Introduction to Software: Kiel Compiler, Proteus. **Interfacing of Led:** Introduction of LED's, Interfacing Circuit Description of LED's, Programming of LED's Interfacing. **Interfacing of Seven Segment Display:** Introduction to 7 Segment Display, Types of 7 Segment Display, Interfacing Circuit Description of 7 Segment Display, Programming of 7 Segment Display Interfacing

Unit 5:
Sensor Interfacing: Introduction to sensing devices, Interfacing of IR Sensors, Interfacing of Temperature Sensor. **Embedded Networking:** I2C Bus Standard, Bluetooth, Zigbee, USB

Text Books:

1. "Embedded Systems: Architecture, Programming and Design" by Raj Kamal
2. "Programming Embedded Systems: With C and GNU Development Tools" by Michael Barr and Anthony Massa

Reference Books:

1. "The 8051 Microcontroller and Embedded Systems: Using Assembly and C" by Muhammad Ali Mazidi, Janice Gillispie Mazidi, and Rolin D. McKinlay
2. **"Embedded Systems: Design, Analysis and Verification" by M.A. Pai**

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the architecture, components, and classifications of embedded systems.
CO2: Develop and debug embedded applications using microcontrollers and programming tools.
CO3: Analyze real-time constraints and apply appropriate design methodologies for embedded solutions.
CO4: Integrate hardware and software components to build functional embedded systems.
CO5: Apply embedded system concepts to solve real-world problems in domains like IoT, automation, and robotics.

BCAI518: Soft Computing						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
1	1	0	35	15	50	1

Teaching Scheme	**Examination Scheme**
Credits: 2	Mid Term Exam: 6 Marks
	Teachers Assessment: 3 Marks
	Attendance: 6 Marks
	End Semester Exam: 35 Marks

Prerequisite: Basic Knowledge of Mathematics and Probability, Fundamentals of Programming and Algorithms

Course Objectives:

1. To introduce the fundamental concepts of soft computing techniques, including fuzzy logic, neural networks, and genetic algorithms.
2. To develop the ability to design and implement intelligent systems using soft computing methods for problem-solving in uncertain and imprecise environments.
3. To explore hybrid systems that integrate multiple soft computing approaches for enhanced performance.
4. To enable students to apply soft computing techniques in various real-world applications such as optimization, pattern recognition, and control systems.

Detailed Syllabus

Unit I: Introduction to Soft Computing and Fuzzy Logic
Overview of Soft Computing and its Need, Differences between Hard and Soft Computing, **Components of Soft Computing:** Fuzzy Logic, Neural Networks, Genetic Algorithms
Introduction to Fuzzy Logic, Fuzzy Sets vs. Crisp Sets, Membership Functions, Fuzzy Rules and Fuzzy Inference System, Fuzzy Decision-Making and Applications, Applications of Fuzzy Logic in Control Systems and Expert Systems

Unit II: Artificial Neural Networks (ANNs)
Biological Neuron vs. Artificial Neuron
Basic Models of ANN: McCulloch-Pitts, Perceptron, Adaline, Madaline, Activation Functions
Learning Rules: Hebbian, Perceptron, Delta Rule
Supervised Learning: Backpropagation Algorithm
Unsupervised Learning: Kohonen Self-Organizing Maps, Competitive Learning
Applications of Neural Networks: Pattern Recognition, Forecasting, and Classification

Unit III: Genetic Algorithms and Hybrid Systems
Introduction to Genetic Algorithms (GAs), Basic Concepts of Evolutionary Computation, Genetic Operators: Selection, Crossover, Mutation, Fitness Function and Selection Techniques, GA Applications in Optimization and Search Problems, Introduction to Hybrid Soft Computing Techniques, Neuro-Fuzzy Systems, Genetic-Fuzzy Systems, GA-NN Hybrid Systems, Case Studies and Applications of Soft Computing in Real-Time Systems

Text and Reference Book

1. S.N. Sivanandam & S.N. Deepa – Principles of Soft Computing, Wiley India
2. Rajasekaran S. & Vijayalakshmi Pai G.A. – Neural Networks, Fuzzy Logic and Genetic Algorithm: Synthesis and Applications, PHI Learning
3. Jang, Sun, and Mizutani – Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Pearson Education
4. Kumar Satish – Neural Networks: A Classroom Approach, Tata McGraw-Hill

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamentals and importance of soft computing techniques such as fuzzy logic, neural networks, and genetic algorithms.
CO2: Design and implement fuzzy inference systems for handling imprecise and uncertain data.
CO3: Develop and train various types of artificial neural networks for classification and prediction tasks.
CO4: Apply genetic algorithms to solve optimization and search problems effectively.
CO5: Integrate multiple soft computing techniques to build hybrid intelligent systems for real-world applications.

BCAI519: Cloud Computing						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
1	1	0	35	15	50	1

Teaching Scheme	**Examination Scheme**
Credits: 2	Mid Term Exam: 6 Marks
	Teachers Assessment: 3 Marks
	Attendance: 6 Marks
	End Semester Exam: 35 Marks

Prerequisite:

1. Fundamentals of Operating Systems and Virtualization.
2. Basic Knowledge of Computer Networks.

Course Objectives:

1. To introduce the fundamental concepts and architecture of cloud computing.
2. To provide knowledge about virtualization and its role in enabling cloud services.
3. To familiarize students with various cloud platforms and technologies.
4. To develop an understanding of cloud security, resource management, and cost optimization strategies

Detailed Syllabus

.

Unit I: Introduction to Cloud Computing:

Definition of Cloud – Evolution of Cloud Computing – Underlying Principles of Parallel and Distributed Computing – Cloud Characteristics – Elasticity in Cloud – On-demand Provisioning.

Unit II: Cloud Enabling Technologies Service Oriented Architecture:

REST and Systems of Systems – Web Services – Publish, Subscribe Model – Basics of Virtualization – Types of Virtualization – Implementation Levels of Virtualization – Virtualization Structures – Tools and Mechanisms – Virtualization of CPU – Memory – I/O Devices –Virtualization Support and Disaster Recovery.

Unit III: Cloud Architecture, Services and Storage:

Layered Cloud Architecture Design – NIST Cloud Computing Reference Architecture – Public, Private and Hybrid Clouds – LAAS – PAAS – SAAS – Architectural Design Challenges – Cloud Storage – Storage-as-a-Service – Advantages of Cloud Storage – Cloud Storage Providers – S3.

Text Books:

1. Kai Hwang, Geoffrey C. Fox, Jack G. Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
2. Rittinghouse, John W., and James F. Ransome, ―Cloud Computing: Implementation, Management and Security, CRC Press, 2017.
3. Rajkumar Buyya, Christian Vecchiola, S. ThamaraiSelvi, ―Mastering Cloud Computing, Tata Mcgraw Hill, 2013.

Reference Books:

1. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing – A Practical Approach, Tata Mcgraw Hill, 2009.
2. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice), O'Reilly, 2009.

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the fundamental concepts, architecture, and service models of cloud computing.
CO2: Explain the role of virtualization in cloud environments and how it supports scalable and flexible computing resources.
CO3: Identify and compare major cloud service providers (e.g., AWS, Azure, GCP) and their core services.
CO4: Apply cloud deployment and management techniques for building, testing, and deploying cloud-based applications.
CO5: Analyze cloud security challenges and implement basic strategies for data protection, access control, and cost management in the cloud.
.

BCAI520: Client Server Computing						
L	**T**	**P**	**Theory**	**Internal**	**Practical**	**Total Marks**
Hours						
1	1	0	35	15	50	1

Teaching Scheme	**Examination Scheme**
Credits: 2	Mid Term Exam: 6 Marks
	Teachers Assessment: 3 Marks
	Attendance: 6 Marks
	End Semester Exam: 35 Marks

Prerequisite: Basic Knowledge of Computer Networks, Fundamentals of Operating Systems

Course Objectives:

1. To introduce the architecture, concepts, and models of client-server computing, including distributed systems and network communication.
2. To enable students to understand and implement client-server applications using appropriate technologies and protocols.
3. To explore various server types and services, such as file servers, database servers, and web servers.
4. To develop skills in managing client-server environments, including security, scalability, and performance considerations.

Detailed Syllabus

Unit I: Fundamentals of Client-Server Computing
Introduction to Client-Server Architecture, Evolution and Characteristics of Client-Server Systems
Components: Clients, Servers, Middleware
Types of Client-Server Architectures: 1-tier, 2-tier, 3-tier, N-tier, Comparison with Centralized and Peer-to-Peer Models, Networking Basics: TCP/IP, Ports, Sockets, Protocols (HTTP, FTP), Role of APIs in Client-Server Communication

Unit II: Technologies and Tools for Client-Server Development
Client-Side Technologies: HTML, CSS, JavaScript, AJAX, Server-Side Technologies: PHP, Java (Servlets/JSP), Python (Flask/Django), Node.js
Web Servers and Application Servers: Apache, Nginx, Tomcat
Database Connectivity: SQL, MySQL/PostgreSQL, ODBC, JDBC, Concepts of Session Management and Cookies, Middleware Technologies: RMI, CORBA, DCOM – Basics and Roles

Unit III: Advanced Topics in Client-Server Systems
Security Aspects: Authentication, Authorization, Firewalls, SSL/TLS
Performance: Caching, Load Balancing, Connection Pooling, Scalability and High Availability in Server Design, Virtualization and Cloud Integration in Client-Server Architecture, Introduction to Microservices and RESTful Services, Monitoring and Maintenance of Client-Server Systems

Text Books and Reference Books:

1. D.T. Dewire – Client/Server Computing, McGraw-Hill Education
2. Robert Orfali, Dan Harkey, Jeri Edwards – The Essential Client/Server Survival Guide, Wiley India
3. James E. Goldman & Phillip T. Rawles – Client/Server Information Systems, Wiley
4. Andrew S. Tanenbaum & Maarten Van Steen – Distributed Systems: Principles and Paradigms, Pearson Education

Course Outcomes:

After completing the course, students will be able to:

CO1: Understand the architecture, components, and working of client-server systems.
CO2: Develop client-server applications using suitable programming languages and frameworks.
CO3: Apply various client-side and server-side technologies for effective communication and data exchange.
CO4: Analyze and implement secure, scalable, and efficient client-server models.
CO5: Evaluate performance issues and implement optimization techniques in client-server environments.