TUMKUR UNIVERSITY

Under Graduate Board of Studies in Computer Science

Curriculum Structure and Syllabus for I and II Semesters

Bachelor of Computer Applications in Artificial Intelligence and Machine Learning

(BCA – AIML)

Choice Based Credit System

2026 - 27 onwards

UG BOS Computer Science Committee Members

1.	**Dr. Basavanna M** Professor & Chairman, Department of Studies in Computer Applications (MCA), Davangere University, Davangere	Chairman
2.	**Dr. Ramani. R** Assistant Professor Department of Computer Science University College of Science Tumkur University, Tumakuru	Member
3.	**Dr. Kusuma Kumari B.M** MCA Coordinator & Assistant Professor Department of Studies and Research in Computer Applications Tumkur University, Tumakuru	Member
4.	**Dr. M. Ravi Kumar** Professor Department of Computer Science Kuvempu University, Shivamogga	External Member
5.	**Dr. Sheetalrani R.K** Assistant Professor Department of Computer Science & Applications Karnataka State Akkamahadevi Women University, Vijayapura.	External Member
6.	**Sri. A.R. Mahesh** Associate Professor Department of Computer Science Government First Grade College, Tumakuru	Member
7.	**Dr. D.K Girija** Associate Professor Department of Computer Science Government First Grade College, Koratagere	Member
8.	**Sri. Nagaraju K V** Associate Professor Department of Computer Science Government First Grade women's College, Tumakuru	Member

Internal Assessment Marks for Theory and Practical

Internal Assessment Marks Allotment for Theory	
Internal Test 1	05
Internal Test 2	05
Assignment	05
Seminar/Quiz	05
TOTAL	**20**

Internal Assessment Marks Allotment for Practical	
Internal Test	05
Record and Attendance	05
TOTAL	**10**

Evaluation Scheme for Lab Examination

Assessment Criteria	
Writing 2 Programs	**10**
Execution of 2 Programs	**20**
Viva	**10**
Total	**40 Marks**

SEMESTER – I

Sl. No	Paper	Title of the Paper	Instruction Hrs. per Week	No. of Credits	Duration of the Exam.	Marks		
						Internal Assessment	Semester End Exam.	Total
1	BCAAIMLT101	Computer Organization and Logic Design	4	4	3 Hrs.	20	80	100
2	BCAAIMLP102	Logic Design Lab	4	2	3 Hrs.	10	40	50
3	BCAAIMLT103	Principles of Programming Languages	4	4	3 Hrs.	20	80	100
4	BCAAIMLP104	MS Office and PPL Lab	4	2	3 Hrs.	10	40	50
5	BCAAIMLT105	Programming in C	4	4	3 Hrs.	20	80	100
6	BCAAIMLP106	Programming in C Lab	4	2	3 Hrs.	10	40	50
TOTAL				**18**				**450**

NOTE:
Theory: 1 Hr. = 1 Credit **Practical: 2 Hrs. = 1 Credit**
BCAAIMLT: BCA Artificial Intelligence and Machine Learning Core Paper Theory
BCAAIMLP: BCA Artificial Intelligence and Machine Learning Core Paper Practical

SEMESTER – II

Sl. No	Paper	Title of the Paper	Instruction Hrs. per Week	No. of Credits	Duration of the Exam.	Marks		
						Internal Assessment	Semester End Exam.	Total
1	BCAAIMLT201	Data Structure Using C	4	4	3 Hrs.	20	80	100
2	BCAAIMLP202	Data Structure Using C Lab	4	2	3 Hrs.	10	40	50
3	BCAAIMLT203	Fundamentals of Artificial Intelligence and Machine Leaning	4	4	3 Hrs.	20	80	100
4	BCAAIMLP204	Prolog Lab	4	2	3 Hrs.	10	40	50
5	BCAAIMLT205	Mathematical Foundations for AIML	4	4	3 Hrs.	20	80	100
6	BCAAIMLP206	Soft Skill Enhancement Lab	4	2	3 Hrs.	10	40	50
TOTAL				**18**				**450**

NOTE:
Theory: 1 Hr. = 1 Credit **Practical: 2 Hrs. = 1 Credit**
BCAAIMLT: BCA Artificial Intelligence and Machine Learning Core Paper Theory
BCAAIMLP: BCA Artificial Intelligence and Machine Learning Core Paper Practical

First Semester			
Subject Name: Computer Organization and Logic Design			
Subject Code	BCAAIMLT101	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To prepare students understand various number systems
2. To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
3. To prepare students to perform the analysis and design of various digital electronic circuits.
4. To acquire basic concepts of computer organization.

COURSE OUTCOMES:

After studying this course, the students would gain enough knowledge on

1. Have a thorough understanding of the fundamental concepts and techniques used in digital electronics.
2. Understand and examine the structure of various number systems and its application in digital design.
3. The ability to understand, analyze and design various combinational and sequential circuits.
4. Ability to identify basic requirements for a design application
5. The ability to understand interfacing techniques of slower devices with faster devices

	Computer Organization	**Hours 15**
Unit 1	Introduction to CPU: Block diagram of computer, characteristics. General register organization of Computer (block diagram, explanation) Bus organization: Address bus, Data bus, Control bus, direction of data flow, Internal data operations in CPU: memory read write, IO read & write. Instruction: definition, format & Classification of instructions. Operation code formats and operand, Instruction fetch, Execute cycle, fetch cycle. Addressing modes: definition, types and examples.	
Unit 2	**Number Representation Techniques**	**Hours 12**
	Introduction, Digital and analog representation, comparison. Number system: Base or radix, Binary, Octal, Decimal, Hexadecimal. Inter conversion techniques Signed and un signed representation (positive and negative representation with 8-bit representation). Binary Arithmetic: Addition, 1s and 2 s compliment, subtraction using 1's and 2's compliment. Binary Codes: Weighted codes BCD, 8421, interconversion applications. Non weighted codes: Excess-3, Gray code. Alphanumeric codes: ASCII, EBCDIC, UNICODE	
Unit 3	**Boolean algebra and Logic gates**	**Hours 13**
	Fundamental Boolean operations and basic logic gates. Derived Gates, Universal Gates NAND and NOR Gates, Universal property. Boolean algebra laws, Demorgan's Theorems. Logic Expression: SOP, POS, canonical representation, SOP and POS to	

	Canonical form conversion. Simplification using Boolean algebra laws Simplification of logic expression using KMAP (without Don't care condition)	
Unit 4	**Combinational logic circuits**	**Hours 10**
	Combinational logic circuits: features, examples, applications. Adder: Half-Adder, Full Adder, implementation of full adder using two Half adders. Subtractor: Half-Subtractor, Full subtractor implementation of full subtractor using two Half subtractors. Encoder: definition, Decimal to BCD encoder, Decoder: definition, BCD to Decimal Decoder Multiplexers: Definition, 4:1 MUX truth table logic diagram. Demultiplexers: Definition, 1:4 DMUX truth table logic diagram. Sequential logic Circuits: features, examples, applications. Latch, concept of clock, Flip-flops conversion of RS flip flop to D flip flop and JK flipflop, JK to T flip flop, Application of Flipflops.	
Unit 5	**Interfacing and Memory Organization**	**Hours 10**
	Introduction, IO interfacing, Data transfer schemes: Synchronous, Asynchronous, DMA. Memory hierarchy: Primary memory, Secondary memory. Memory organization: Cache memory, virtual memory, memory management hardware Memory stack. Memory address decoding (3bit address).	

TEXT BOOKS:

1. Computer System Architecture by M Morris Mano.

REFERENCE BOOKS:

1. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012
2. Digital fundamental by Thomas L. Floyd
3. Digital Electronics by A K Maini
4. Computer Organization and architecture by William Stallings

First Semester			
Subject Name: Logic Design Lab			
Subject Code	BCAAIMLP102	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Verification of truth tables for AND, OR, NOT gates.
2. Verification of truth tables for EXOR and EXNOR gates
3. Verification of truth tables for NAND and NOR gates.
4. Realization of AND, OR, NOT gates using only NAND gates.
5. Realization of AND, OR, NOT gates using only NOR gates.
6. Verification truth table for Half adder using NAND gates.
7. Verification truth table for Full adder using NAND gates.
8. Verification of truth table for Full subtractor using XOR gates and Basic gates
9. Verification truth table for Half subtractor using NAND gates.
10. Verification of truth table for Decimal to BCD Encoder

PART B:

1. Verification of truth table for BCD to Seven segment display decoder.
2. Verification of Distributive property.
3. Verification of truth table for BINARY TO GRAY conversion using XOR gates
4. Verification of truth table for GRAY TO BINARY conversion using XOR gates
5. Verification of truth table for SR and D Flip-Flop.
6. Verification of truth table for JK and T Flip-Flop.
7. Design of logic circuit for simple 2 variable SOP expression.
8. Design of logic circuit for simple 2 variable POS expression.
9. Design of logic circuit of De-Morgans' Theorem and realize.
10. Simplification of given expression using KMAP and designing the logic circuit. (SOP only)

Practical can be performed using any open-source simulator (like Logisim)
(Download it from https://sourceforge.net/projects/circuit/)

First Semester			
Subject Name: Principles of Programming Languages			
Subject Code	BCAAIMLT103	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. For the beginner students learning by using text based is difficult to programming concepts.
2. By using flowcharts, Students can concentrate on programming concepts rather than all the nuances of a typical programming language

COURSE OUTCOMES:

After studying this course, the students able to,

1. To introduce the various programming paradigms.
2. To understand the evolution of programming languages.
3. To understand the concepts of OO languages, functional languages, logical, scripting Languages and modern programming languages.
4. To introduce the notations to describe the syntax and semantics of programming languages.

Unit 1	**Preliminary Concepts**	**Hours 15**
	Reasons for studying concepts of programming languages, programming domains, language evaluation criteria, influences on language design, language categories, language design trade-offs, implementation methods, programming environments. Syntax and Semantics: General problem of describing syntax, formal methods of describing syntax, attribute grammars, describing the meanings of programs, Main Programming Structures, Algorithm, Flow Charts & Pseudocode, documenting a Program, understand programming logic through Flogorithm	
Unit 2	**Programming Language**	**Hours 15**
	Introduction, Key Concepts and overview of: Functional Programming Languages, Logic Programming Language, Scripting Languages. Object-Oriented Programming: Design issues for OOP, Using the Unified Modelling Language to Design Classes, Inheritance, Polymorphism GUI Programming Languages: GUI Applications, Graphical User Interfaces, Designing the User Interface and Event Handler for a GUI Program.	
Unit 3	**Basic concepts of Program**	**Hours 10**
	Variables and scopes: Introduction, names, variables, concept of binding, scope, scope and lifetime, referencing environments, named constants, Operators Data types: Introduction, primitive, character, string types, user defined ordinal types, array, associative arrays, record, tuple types, list types, union types, pointer and reference types, type checking, strong typing, type equivalence.	
Unit 4	**Statements and Control Structures**	**Hours 10**
	Expressions and Statements: Arithmetic expressions, overloaded operators, type conversions, relational and Boolean expressions, short- circuit evaluation, assignment statements, mixed mode assignment. Control Structures – introduction, selection statements, iterative statements,	

	unconditional branching, guarded commands.	
Unit 5	**Sub Programs**	**Hours 10**
	Introduction to Sub-Programs (Functions): Defining and Calling a Module, Local Variables, Passing Arguments to Modules, Global and Local Variables. Calling subprograms indirectly, implementing subprograms, General semantics of calls and returns, implementing subprograms with stack-dynamic local variables, nested subprograms, blocks. Concept of Abstract Data types. Library Functions, Introduction to Recursion and Problem Solving with Recursion. Introduction to Menu-Driven Programs: Using a Loop to Repeat the Menu, Modularizing a Menu-Driven Program, Multiple-Level Menus	

TEXT BOOKS:

1. Concepts of Programming Languages, 11E, Robert Sebesta, Pearson, Global Edition.

REFERENCE BOOKS:

1. Starting out with Programming Logic & Design, 3E, Tony Gaddis, Pearson Education
2. Programming language design concepts, Watt, David A. John Wiley & Sons Ltd.
3. The Art of Programming through Flowcharts & Algorithms, 2E, A. B. Chaudhuri, Firewall Media

First Semester			
Subject Name: MS Office and PPL Lab			
Subject Code	BCAAIMLP104	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Design a visiting card for Managing Director of a company as per the following specification.
   - Size of visiting card is 3½×2
   - Name of the company with big font
   - Phone number, Fax number and E-mail address with appropriate symbols.
   - Office and Residence address separated by a line.

2. Create a table with following columns and display the result in separate cells for the following
   - Emp Name, Basic pay, DA, HRA, Total salary.
   - Sort all the employees in ascending order with the name as the key
   - Calculate the total salary of the employee
   - Calculate the Grand total salary of the employee
   - Finding highest salary and
   - Find lowest salary

3. Prepare an advertisement to a company requiring software professional with the following
   - Attractive page border
   - Design the name of the company using WordArt
   - Use at least one clipart.
   - Give details of the company (use bullets etc.)
   - Give details of the Vacancies in each category of employee's (Business manager, Software engineers, System administrators, Programmers, Data entry operators) qualification required.

4. Create two pages of curriculum vitae of a graduate with the following specifications
   - Table to show qualifications with proper headings
   - Appropriate left and right margins
   - Format ½ page using two-column approach about yourself
   - Name on each page at the top right side
   - Page no. in the footer on the right side.

5. Create a letter as the main document and create 10 records for the 10 persons use mail merge to create letter for selected persons among 10.

6. Create a excel spreads sheet to calculate the net pay of the employees following the conditions below.

	A	B	C	D	E	F	G	H	I
1	EMP ID	EMP NAME	BASIC	DA	HRA	GPF	INCOME TAX	GROSS SAL	NET SAL
2									

- DA: -56% of the basic pay if Basic pay is greater than 20000 or else 44%.
- HRA: -15% of the Basic pay subject to maximum of Rs. 4000.
- GPF: -10% of the basic pay.
- INCOMETAX: - 10% of basic if Basic pay is greater than 20000.
- Find who is getting highest salary & who is get lowest salary?

7. The ABC Company shows the sales of different product for 5 years. Create BAR Graph, 3D and Pie chart for the following.

A	B	C	D	E	F
SL.NO.	YEAR	PROD1	PROD2	PROD3	PROD4
1	2019	1000	800	90	1000
2	2020	800	90	70	1100
3	2021	1200	190	100	900
4	2022	600	260	58	1400
5	2023	1800	510	80	800

8. Create a suitable examination data excel sheet and find the sum of the marks (total) of each student and respective, class secured by the student.

- Pass: if marks in each subject>=35
- Distinction: if average>=75
- First class: If average>=60 but <75
- Second Class: if average >=50 but less than 60
- Third class: if average>=35 but less than 50
- Fail: if marks in any subject<35

9. Enter the following data into the sheet.

NAME	DEPARTMENT	SALARY
Anusha	Accounts	30000
Ramesh	Marketing	20000
Tejaswi	Engineering	44000
Harika	Accounts	35000
Poornima	Engineering	56000
Vijay	Marketing	22000
Prasad	Accounts	48000
Swetha	Engineering	60000
Rajesh	Marketing	19000

- Extract records for department Accounts and Salary > 10000
- Sort the data by salary with the department using “sort commands”.
- Calculate total salary for each department using Subtotals

10. Create a presentation with slide transitions and animation effects.

PART B:

Design Flowchart and submit to generate pseudo code and run the program for the following problems (Any 12):

1. Read radius to calculate and display area of circle.
2. Find maximum of three number
3. Count no of digits in a number.
4. Reverse a given number.
5. Print odd number from given numbers.
6. Find minimum, maximum and average score of labs.
7. Print even number between 2 to n
8. Determine the give year is leap or not
9. Finding Duplicates in an Array.
10. Defining and calling a function.
11. To perform bubble sort.
12. Print triangle pattern
13. To search an element using linear search.
14. Solve quadratic equation.
15. Reverse a string.

First Semester			
Subject Name: Programming in C			
Subject Code	BCAAIMLT105	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. The course aims to provide exposure to problem-solving through programming.
2. It aims to train the student to the basic concepts of the C-programming language.
3. This course involves a lab component which is designed to give the student hands-on experience with the concepts.

COURSE OUTCOMES:

After completing this course satisfactorily, a student will be able to:

1. Confidently operate Desktop Computers to carry out computational tasks
2. Understand working of Hardware and Software and the importance of operating systems
3. Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
4. Read, understand and trace the execution of programs written in C language
5. Write the C code for a given problem
6. Perform input and output operations using programs in C
7. Write programs that perform operations on arrays.

Unit 1	**C Programming Basic**	**Hours 10**
	Introduction and Overview: 'C' History and Background, Example 'C' Program, C Program Structure. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - printf and scanf, control stings and escape sequences, output specifications with printf functions; Unformatted I/O functions - getchar, putchar, gets and puts functions.	
Unit 2	**Operators & Expressions**	**Hours 15**
	Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, Bitwise operators, Conditional operator, Special operators, Operator Precedence and Associatively; Evaluation of arithmetic expressions; Type conversion. Control Structures: Decision making Statements - Simple if, if_else, nested if_else, else_if ladder, Switch Case, goto, break & continue statements; Looping Statements - Entry controlled and exit controlled statements, while, do-while, for loops, Nested loops.	

Unit 3	**Derived data and Strings**	**Hours 12**
	Derived data types in C: Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - strlen, strcmp, strrev, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc.	
Unit 4	**Functions and Structures**	**Hours 13**
	User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure member's initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	
Unit 5	**Pointers**	**Hours 10**
	Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Ponter and function; Advantages and disadvantages of using pointers.	

TEXT BOOKS:

1. E. Balaguruswamy: Programming in ANSI C (TMH)
2. C: The Complete Reference, By Herbert Schildt.
3. C Programming Language, By Brain W. Kernighan
4. Kernighan & Ritchie: The C Programming Language (PHI)

REFERENCE BOOKS:

1. P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
2. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
3. V. Rajaraman: Programming in C (PHI – EEE)
4. S. Byron Gottfried: Programming with C (TMH)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

First Semester			
Subject Name: Programming in C Lab			
Subject Code	BCAAIMLP106	**CIE Marks**	10
No of Hours/Week:	4	**SEE Marks**	40
Total Hours:	60	**Credits**	2

LIST OF PRACTICAL PROGRAMS

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
   a. Familiarization of Computer Hardware Parts
   b. Basic Computer Operations and Maintenance.
   c. Do's and Don'ts, Safety Guidelines in Computer Lab

2. Familiarization of Basic Software – Operating System, Basic DOS Commands, Word Processors, Spread Sheets, PowerPoint, Internet Browsers, Integrated Development Environment (IDE) with Examples.

3. Type Program Code preferably by using DevC++ or Codeblock or VisualCode, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

PART A:

1. Write a C Program to read radius of a circle and to find area and circumference
2. Write a C Program to read three numbers and find the biggest of three
3. Write a C Program to demonstrate library functions in *math.h*
4. Write a C Program to check for prime
5. Write a C Program to generate n primes
6. Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
10. Write a C program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Write a C Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

PART B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.

3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime( )* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of n students.
12. Write a C Program to demonstrate the difference between structure & union.

Second Semester			
Subject Name: Data Structure Using C			
Subject Code	BCAAIMLT201	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To explore and understand the concepts of Data Structures and its significance in programming.
2. To teach efficient storage mechanisms of data for an easy access
3. Provide and holistic approach to design, use and implement abstract data types.
4. Understand the commonly used data structures and various forms of its implementation for different applications using C
5. To improve the logical ability.

COURSE OUTCOMES:

1. Learn about Data structures, its types and significance in computing.
2. Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Explore about Abstract Data types and its implementation
4. Ability to program various applications using different data structure in C

	Introduction to Data Structure	**Hours 10**
Unit 1	**Data structure –** Definition, Abstract data type, Classification of Data Structures, Operations on Data Structures-primitive and non-primitive. **String Processing**: Definition, Storing Stings, String as ADT, String operations – string length, copy, compare, pattern matching using user defined function and pointer. **Memory allocation**: Definition, types-static and dynamic, difference between static and dynamic. Memory management functions- malloc, calloc, realloc and free.	
	Arrays	**Hours 10**
Unit 2	**Arrays:** Definition, Linear arrays, memory representation of Linear Array, Accessing the Elements of an Array, Calculating the Address of Array Elements. Array operations - Inserting, deleting and merging. **Sorting**: Bubble sort, Insertion sort, Selection sort. **Searching**: Linear Search and Binary search. **Multidimensional array-** Declaration and memory representation, Sparse matrices.	
	Linked Lists	**Hours 15**
Unit 3	**Linked list:** Definition, Types. Memory representation of Singly linked list, operations- create, display, insert and delete. Doubly liked list- definition and memory representation. Circular linked list-definition and memory representation. Advantages and disadvantages of SLL, DLL and CLL. Garbage collection.	

Unit 4	**Stacks and Queues**	**Hours 15**
	Stacks: Definition, Array representation of stacks, Linked representation of stacks, Operations on stack. Expressions: Polish Notation, Application of Stacks. Algorithm – Convert infix to postfix expression and evaluate postfix expression. **Queues** – Definition, Array representation of queue, Linked list representation of queues. Types of queues: Simple queue, Circular queue, Double ended queue, Priority queue, Operations on Queues, Applications of queues.	
Unit 5	**Graphs and Trees**	**Hours 10**
	Graphs: Graph theory terminology: pendent vertex, isolated vertex, degree of vertex, path, walk, self-loop, cycle, Sequential representation of Graphs: Adjacency matrix. **Tree –** Definitions, Types of trees, memory representation of binary tree, tree traversing techniques, Binary Search Trees- definition and construction of binary search tree. AVL Tree- definition. Threaded Binary tree – Definition.	

TEXT BOOK:

1. Data Structure using C by Reema Thereja, Oxford University Press.

REFERENCE BOOKS:

1. Data Structure by S. Lipschutz - (Schaum Series)
2. Introduction to Data Structure in C by: A.N.Kamthane; Pearson Education
3. Alfred V Aho, John E Hopcroft and Jeffery D Ullman, “Data Structures and Algorithms”, Pearson Education.

Second Semester			
Subject Name: Data Structure Using C Lab			
Subject Code	BCAAIMLP202	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Develop a Program in C for the operations on Strings like finding the string of length, copying two strings, comparing of two string and pattern matching & replacing. Support the program with functions for each of the above operations. Don't use Built-in functions
2. Write a C program to read name and roll number of n number of students from user and store them in a file.
3. Write a C Program to implement dynamic array, find smallest and largest element of the array.
4. Write a C Program read, display and to find the trace of a square matrix
5. Write a C Program to read, display and add two m x n matrices using functions
6. Write a C Program to read, display and multiply two m x n matrices using functions
7. Write a C Program to read the names of cities and arrange them alphabetically.
8. Write a C Program to search an element using linear search technique.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a program to implement merge sort.

PART B:

1. Program to implement linear linked list to perform insert and delete operations on it.
2. Write a C Program to implement Stack and its different operations.
3. Write a C Program to convert an infix expression to postfix.
4. Write a C Program to evaluate a postfix infix expression.
5. Write a C Program to implement simple queue and its different operations.
6. Write a program to implement circular queue using array.
7. Program to create and display different traversal of a binary tree.
8. Write a program to implement BFS.
9. Write a program to implement DFS.
10. Write a program to implement AVL Tree.

Second Semester			
Subject Name: Fundamentals of Artificial Intelligence and Machine Learning			
Subject Code	BCAAIMLT203	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	4

COURSE OBJECTIVES:

1. To understand the basic concepts, history, and applications of AI.
2. To understand how AI systems act and solve problems.
3. To understand how knowledge is stored and used in AI systems.
4. To introduce students to fundamental machine learning concepts.
5. To introduce modern AI technologies at a basic level.
6. To understand the basic concepts of machine learning.
7. To introduce prolog programming language.

COURSE OUTOMES(COs):

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

1. Explain fundamental concepts, history, goals, and applications of Artificial Intelligence.
2. Identify and classify different types of AI and intelligent agents used in problem-solving environments.
3. Apply basic problem-solving and search techniques to represent and solve simple AI problems.
4. Describe knowledge representation methods and reasoning mechanisms used in AI systems.
5. Demonstrate understanding of Machine Learning and Artificial Neural Network concepts at a basic level.
6. Use Generative AI tools to create text, images, and simple code for academic and practical purposes.
7. Design effective prompts and explain the working principles of Large Language Models.
8. Explain the architecture and working of Agentic AI systems and their components.
9. Analyze ethical, legal, and social issues related to AI and practice responsible AI usage.
10. Identify emerging trends and career opportunities in Artificial Intelligence.
11. Explain the fundamental concepts of machine learning.
12. Explain the prolog programming language.

UNIT 1	**Introduction to Artificial Intelligence**	**Hours 6**
	Introduction to AI: Definition of AI, Goals of AI, Characteristics of Intelligent Systems **History of AI:** Early developments, Growth of AI, Modern AI era **Types of AI:** Narrow AI, General AI, Super AI (concept only) **Applications of AI: -** Healthcare, Education, Banking, E-commerce, Chatbots (e.g., ChatGPT) **AI vs Machine Learning vs Deep Learning (Basic Difference)**	
UNIT 2	**Intelligent Agents and Problem Solving**	**Hours 8**
	Intelligent Agents: Definition of Agent, Agent and Environment, Types of	

	Agents- Simple Reflex Agent, Model-Based Agent, Goal-Based Agent, Utility-Based Agent **Problem Solving in AI:** Problem formulation, State space representation **Search Techniques (Basic Concepts Only) :** Breadth First Search (BFS), Depth First Search (DFS), Introduction to Heuristics	
UNIT 3	**Knowledge Representation and Reasoning**	**Hours 14**
	Knowledge Representation: What is Knowledge? Propositional Logic (basic), Predicate Logic (basic idea), Semantic Networks, Frames **Reasoning in AI :** Forward Chaining, Backward Chaining **Introduction to Advanced AI Areas:** **Introduction to Expert Systems:** Components of Expert System, Applications, Example: MYCIN (basic discussion) **Artificial Neural Networks (Basic Idea):** Biological neuron vs Artificial neuron, Perceptron (concept only) **Natural Language Processing (NLP):** Text processing, Chatbots, Example: Siri **Computer Vision:** Image recognition, Face detection **Robotics:** AI in Robotics, Autonomous systems **Future of AI:** Career opportunities, Ethical issues in AI	
UNIT 4	**Basics of Machine Learning (*Only introductory level*)**	**Hours:12**
	Introduction to Machine Learning: Definition, Importance **Types of Learning:** Supervised Learning, Unsupervised Learning, Reinforcement Learning (concept only) **Basic Algorithms (Conceptual Understanding):** Linear Regression, k-Nearest Neighbor (KNN), Decision Tree, K-Means Clustering **Basic Concepts:** Training and Testing Data, Accuracy, Overfitting and Underfitting	
UNIT 5	**Introduction to Prolog**	**Hours 20**
	An Overview of Prolog: An example program: defining family relations, Extending the example program by rules, A recursive rule definition, How Prolog answers questions, Declarative and procedural meaning of programs. **Syntax and Meaning of Prolog Programs**: Data objects, Matching, Declarative meaning of Prolog programs, Procedural meaning, Example: monkey and banana **Lists, Operators, Arithmetic:** Representation of lists, Some operations on lists, Operator notation, Arithmetic. **Using Structures: Example Programs:** Retrieving structured information from a database, Doing data abstraction, Simulating a non-deterministic automaton, Travel planning, The eight queens problem. **Programming Style and Technique**: General principles of good programming How to think about Prolog programs Programming style Debugging Efficiency. **Basic Problem-Solving Strategies**: Introductory concepts and examples Depth-first search strategy Breadth-first search strategy	

TEXT BOOKS:

1. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson.
2. Elaine Rich, Kevin Knight, Shivashankar B. Nair, *Artificial Intelligence*, Tata McGraw-Hill.
3. PROLOG PROGRAMMING FOR ARTIFICIAL INTELLIGE, lvan Bratko E. Kardelj University . J. Stefan Institute Yugoslavi, ADDISON-WESLEY PUBLISHING COMPANY.

REFERENCE BOOKS:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press.
2. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly.
3. Joseph Giarratano & Gary Riley, *Expert Systems: Principles and Programming*, Cengage.

Second Semester			
Subject Name: Prolog Lab			
Subject Code	BCAAIMLP204	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Installation and study of prolog.
2. Write a program in prolog to add two numbers.
3. Write a program in prolog to simple fact for the statements.
4. Write a program in prolog to implement simple arithmetic.
5. Write a program in prolog to implement Factorial of a given number.
6. Write a program in prolog to implement Fibonacci of a given number.
7. Write a program in prolog to predicates one converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
8. Write a program in prolog categorize animal characteristics.
9. Write a program in prolog to show how integer variable is used in prolog program
10. Write a program in prolog to implement simple facts and Queries.

PART B:

1. Write a program in prolog to demonstrate family relationship.
2. Write a program in prolog to solve Monkey banana.
3. Write a program to reverse the list.
4. Write a program to count number of elements in a list.
5. Write a program in prolog to solve Tower of Hanoi.
6. Write a program in prolog to solve travelling salesman problem.
7. Write a program in prolog to solve 4 queen's problem.
8. Write a program in prolog to solve any problem using depth first search.
9. Write a program in prolog to solve any problem using breadth first search.
10. Write a program in prolog to solve 8 Puzzle problems.

Second Semester			
Subject Name: Mathematical Foundations for AIML			
Subject Code	BCAAIMLT205	CIE Marks	20
No of Hours/Week:	4	SEE Marks	80
Total Hours:	60	Credits	04

COURSE OBJECTIVES:

1. The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete and basic probability.
2. These concepts are useful to study or describe objects or problems in computer algorithms and programming languages.

COURSE OUTCOMES:

After studying this course, the students would gain enough knowledge on

1. Verify the correctness of an argument using propositional and predicate logic and truth tables.
2. Solve problems on Sets, Relations and functions.
3. Construct the proofs by using proof by mathematical induction, principles of counting and application of permutation and combination.
4. Understand the terminologies of basic probability, two types of random variables and their probability functions
5. Observe and analyze the behavior of various discrete and continuous probability distributions
6. Enable study of probability concept required for Computer learners

Unit 1	**Mathematics Logic**	**Hours 15**
	Definition of proposition, proposition variables, negation of statements, logical connectives and truth tables, predicates and quantifiers, Conditional statement, contra-positive and converse, bi-conditional, tautology, contradiction, logical equivalence, properties of proposition operation-commutative, associative, distributive, idempotent negation. Introduction to proofs. Methods of proof.	
Unit 2	**Set Theory**	**Hours 8**
	Definition of a set, sub-set with examples, Venn diagrams, types of sets-equal sets, null set, disjoint sets, finite set, infinite set, power set, cardinality of set. Operations on sets. Definition of a relation with examples, types of relations. Definition of a function with examples, types.	
Unit 3	**Matrices**	**Hours 13**
	Definition of matrix and order of matrix, types of matrices, operations on matrices, inverse of a matrix, determinant of matrix, properties of determinant, applications of determinants and matrices. Finding rank of a matrix, normal form, echelon form, Cayley Hamilton theorem, Eigen values and Eigen vectors.	
Unit 4	**Basic Probability**	**Hours 12**
	Probability: Introduction, random experiments, sample space, events and algebra of events. Definitions of Probability – classical, statistical, and axiomatic. Conditional	

	Probability, laws of addition and multiplication, independent events, Bayes' rule, Bernoulli trials.	
Unit 5	**Random Variables**	**Hours 12**
	Random variables: Discrete and Continuous random variables, Probability density function, Probability distribution of random variables, Expected value, Variance. Distributions: Discrete distributions: Uniform, Binomial, Poisson, Bernoulli, Continuous distributions: uniform distributions, exponential, Normal distribution state all the properties and its application.	

TEXT BOOKS:

1. Kenneth H Rosen: Discrete Mathematics and its Applications, McGraw Hill publications, 7th edition, 2007.

2. J. P. Tremblay and R.P. Manohar: Discrete Mathematical Structures with applications to Computer Science, Mc Graw Hill Ed. Inc. 2002.

3. S. C. Gupta, V. K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand & Sons Educational Publishers, New Delhi, 2020.

REFERENCE BOOKS:

1. Ralph P. Grimaldi, Discrete and Combinatorial Mathematics, 5thEdition, Pearson Education, 2004.

2. C. L. Liu and D. P. Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, McGraw Hill, Third Edition, 2012.

3. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall.

4. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India.

5. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, Wiley.

Second Semester			
Subject Name: Soft Skill Enhancement Lab			
Subject Code	BCAAIMLP206	CIE Marks	10
No of Hours/Week:	4	SEE Marks	40
Total Hours:	60	Credits	2

LIST OF PRACTICAL PROGRAMS

PART A:

1. Basics of Communication Skills and Listening comprehension.
2. Group Discussion – dynamics of group discussion, Lateral thinking, Brainstorming.
3. Personality Development: Decision-Making, Problem Solving, Goal Setting, Time Management & Positive Thinking.
4. Writing Skills: Letter writing, Essays for Competitive examinations.
5. Resume writing – structure and presentation, planning, defining the career objective.
6. Body Language-To reveals your inner self and personality.
7. Meetings- making meeting effective, chairing a meeting, decision-making, seeking opinions, interrupting and handling interruptions, clarifications, closure, negotiation skills.
8. Interview Skills – concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele and video-conferencing.

PART B:

1. Mock interview.
2. Official Mail id Creation
3. Micro Presentation
4. NAD registration Step by Step
5. Use of word processing, presentation tools for communication and presentation skills.
6. Use of Google forms, drive for collaborative work.
7. Use of spreadsheet for data interpretation and data analysis.
8. Using netiquettes in online mode of communication using Zoom / Google Meet / MS-Teams etc.

TEXT BOOKS:

1. Personality Development and SOFT SKILLS, BARUN K. MITRA Oxford University Press.
2. M. Ashraf Rizvi, “Effective Technical Communication”, Tata McGraw-Hill Publishing Company Ltd. 2005.

REFERENCE BOOKS:

1. Andrea J. Rutherford, “Basic Communication Skills for Technology”, 2nd Edition, Pearson Education, 2007.
2. Meenakshi Raman & Sangeeta Sharma, “Technical Communication”, Oxford University Press, 2011. DELTA’s key to the Next Generation TOEFL Test: “Advanced Skill Practice,” New Age International (P) Ltd., Publishers, New Delhi.

Question Paper Pattern for Semester End Examination (SEE)
BCA Artificial Intelligence and Machine Learning
(Common for I and II Semester)
SUBJECT NAME

Time: 3 Hours **Max. Marks: 80**

Instruction to Candidate: Answer all the Sections

SECTION A

I. Answer any ten of the following questions (10X2 = 20)

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.

SECTION B

II. Answer any five of the following questions (5X5 = 25)

13.
14.
15.
16.
17.
18.
19.

SECTION C

III. Answer any five of the following questions (5X7 = 35)
20.
21.
22.
23.
24.
25.
26.