University of Engineering and Management
Institute of Engineering & Management,SaltLake Campus
Institute of Engineering & Management,NewTown Campus
University of Engineering &Management,Jaipur

1st Semester Syllabus for MCA Admission Batch 2024

Index:

SyllabusStructure:

THEORY							
SL. NO.	**SUBJECT CODE**	**SUBJECT NAME**	**CONTACTS (PERIODS / WEEK)**				**CREDITS**
			L	**T**	**P**	**TOTAL**	
1	MCA101	Computer Organization and Architecture	3	1	0	4	3
2	MCA102	Computer Programming with C	3	1	0	4	4
3	MCA103	Data Structures with C	3	1	0	4	4
4	MCA104	Discrete Mathematical Structure	3	1	0	4	3
5	MCA105	Business English and Communication	3	1	0	4	3
6	MCAESP101	Essential Studies for Professionals - I	2	0	0	2	0.5
Total of Theory						22	17.5
PRACTICAL							
7	MCA191	Micro Programming and Architecture Laboratory	0	0	3	3	3
8	MCA192	C Programming Laboratory	0	0	3	3	3
9	MCA193	Data Structures with C Laboratory	0	0	3	3	3
Total of Practical						9	9
SESSIONAL							
10	MCASDP181	Skill Development for Professionals - I	2	0	0	2	0.5
11	MAR	Mandatory Additional Requirements	0	0	0	0	0
12	IFC	Industry and Foreign Certification	0	0	0	0	0
13	MOOCS	Massive Open Online Courses	0	0	0	0	0
Total of Sessional						2	0.5
Total of Semester						33	27

Subject Name: Computer Organisation and Architecture **Credit: 4**

LectureHours:40 **Subject Code: MCA101**

Name of the Course: **Computer Organization and Architecture**	
CourseCode:MCA101 &MCA191	**Semester:1st**
Duration:40Hrs.	**Maximum Marks:100**
TeachingScheme	**Examination Scheme**
Theory:3	End Semester Exam:100
Tutorial: 1	Continuous Assessment:100
Practical:2	Practical Sessional internal continuous evaluation:100
Credit:4+2	Practical Sessional external examination:100
Aim:	
Sl.No.	
1	To have a thorough understanding of the basic structure and operation of a digital computer.
2	To study the different ways of communicating with I/O devices and standard I/O interfaces.
3	To learn the architecture and assembly language programming of 8085 microprocessor.
Objective:	

Sl.No.	
1	Understanding Logic gates, flip flops and counter.
2	Clear Understanding of Computer Architecture.
3	Clear Understanding Pipeline processing ,RISC and CISC architectures.
4	Develop abase for advance micro-processors.
Pre-Requisite:	
Sl.No.	
1.	Proficiencyinbasic Digital Electronics
CourseOutcome:	
1.	Summarize the fundamental components of basic computer system and its organization.
2.	Apply arithmetic and logical microoperations of binary number system.
3.	Analyze control unit design and concept of pipelining.
4.	Classify memory hierarchy and examine numerical problem based on it.
RelevantLinks:	
COA Linkedin Learning Link	COA Coursera Link COA NPTEL Link

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO1	3	2	2	3	-	-	-	-	-	2	-	2	-	-	-
CO2	3	2	2	2	-	-	1	-	-	2	-	2	-	-	-
CO3	2	2	3	3	-	-	1	-	-	2	-	2	-	-	-
CO4	3	2	3	2	-	-	-	-	-	2	-	2	-	-	-

Module number	**Topic**	**Sub-topics**	**MappingwithIndustry and International Academia**	**Lecture Hours**	**CorrespondingLabAssignment**
1	**STRUCTURE OF COMPUTERS and COMPUTER ARITHMETIC**	Computer types, Functional units, Basic operational concepts, VonNeumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Data representation, Fixed and Floating point, Error detection and correction codes Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations.	***International Academia:*** https://web.stanford.edu/dept/registrar/bulletin_past/bulletin02-03/pdf/CompSci.pdf ***AICTE-prescribed syllabus:*** ***IndustryMapping:***The concepts delivered are in sync with the industry standards	4	1. Write a VHDL code to study and perform about logic gates. 2. Write a VHDL code to study and perform about De'Morgan's Theorem. Write a VHDL codeto study and perform about NAND and NOR as a universal gates. 3. Write a VHDL codeto design and implement circuit that converts binary code to gray code.

2	**BASIC COMPUTER ORGANIZATION AND DESIGN**	Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input-Output and interrupt. Central processing unit: Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC vs RISC	***International Academia:*** https://web.stanford.edu/dept/registrar/bulletin_past/bulletin02-03/pdf/CompSci.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	8	4. Write a VHDL codeto study and perform about Half Adder and full Adder. 5. Write a VHDL codeto study and perform about Half substractor and full substractor. 6. Write a VHDL codeto design 3-bit odd/even parity generator and checker. 7. Write a VHDL codeto study and perform about R-S and D flip flop. 8. Write a VHDL codeto study and perform about J-K and T flip flop. 9. Write a VHDL codeto study and perform about Master slave JK flip flop.
3	**REGISTER TRANSFER AND MICRO-OPERATIONS and . MICRO-PROGRAMMED CONTROL**:	Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit, Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit.	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	8	10. Write a VHDL codeto realize Boolean functions using multiplexer. 11. Write a VHDL codeto study and perform about Decoder and Demultiplexer. 12. Write a VHDL codeto study the use of decoder for BCD to seven segment LED display. 13. Write a VHDL codeto study universal shift register

4	**MEMORY SYSTEM:**	Memory Hierarchy, Semiconductor Memories, RAM(Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID.	***International Academia:*** https://web.stanford.edu/dept/registrar/bulletin_past/bulletin02-03/pdf/CompSci.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aict https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:*** The concepts delivered are in sync with the industry standards	7	
5	**INPUT OUTPUT:**	I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA.	***International Academia:*** https://web.stanford.edu/dept/registrar/bulletin_past/bulletin02-03/pdf/CompSci.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:*** The concepts delivered are in sync with the industry standards	7	
6	**MULTIPROCESSORS**	Characteristics of multiprocessors, Interconnection structures, Inter Processor Arbitration, Inter processor	***International Academia:*** https://web.stanford.edu/dept/registrar/bulletin_past/bulletin02-03/pdf/CompS		

		Communication and Synchronization, Cache Coherence.	ci.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:*** The concepts delivered are in sync with the industry standards		

ListofBooksTextBooks:			
Nameof Author	**Title oftheBook**	**Edition/ISSN/ISBN**	**NameofthePublisher**
M. Moris Mano	Computer System Architecture	3rd Ed	Pearson/PHI
ReferenceBooks: **1. Carl Hamacher, ZvonksVranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India.**			

Subject Name: Computer Programming with C Credit: 4 Lecture Hours:40 Subject Code: MCA102

Name of the Course: **Computer Programming with C**	
CourseCode:MCA102 &MCA192	**Semester:1st**
Duration:40Hrs.	**Maximum Marks:100**
Teaching Scheme	**Examination Scheme**
Theory:3	End Semester Exam:100
Tutorial: 1	Continuous Assessment:100
Practical:2	Practical Sessional Internal continuous evaluation:100
Credit:4+2	Practical Sessional external examination:100
Aim:	
Sl.No.	
1	T ogain Knowledge of Various aspects of algorithm development
2	To enhance the ability to identify the qualities of a good solution
3	To implement learned algorithm design techniques and data structures to solve problems.
Objective:	
Sl.No.	
1	The fundamental design,analysis and implementation of basic data structures.

2	Basic concepts in the specification and analysis of programs.
3	Principles for good program design,especially the uses of data abstraction.
4	Significance of algorithms in the computer field
Pre-Requisite:	
Sl.No.	
1.	Proficiency in one high-level programming language
CourseOutcome:	
1.	Apply fundamental programming concepts like control flow, functions, and data structures to solve computational problems using C.
2.	Analyze and design algorithms for various problems using C programming paradigms.
3.	Implement efficient and modular C programs with proper memory management techniques.
4.	Understand and utilize advanced C features like pointers, file handling, and preprocessor directives for building complex applications.
RelevantLinks:	
CStudyMaterial	C NPTELLINK C CourseraLink CLinkedInLearningLink

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	2	3	1	0	1	2	1	1	2	3	2	2
CO2	2	3	3	3	2	1	0	1	2	1	1	2	3	2	2
CO3	3	2	3	2	3	1	0	1	2	1	1	2	3	2	2

CO4	2	2	3	2	3	1	0	1	2	1	1	2	3	2	2

Module number	Topic	Sub-topics	MappingwithIndustry and International Academia	Lecture Hours	CorrespondingLabAssignment
1	Basics of 'C' Programming	1.1 Fundamentals of algorithms: Notion of algorithm, Notations used for assignment statements and basic control structures. 1.2 Introduction to 'C': General structure of ' C' program, Header file, 'main ()' function. 1.3 Fundamental constructs of 'C': Character set, tokens, keywords, Identifiers, Constants - number constants, character constants, string constants, Variables. Data types in 'C': Declaring variables, data type conversion. 1.4 Basic Input and Output functions: input and output statements using printf(), scanf() functions. 1.5 Assignments and expressions: simple assignment statements, arithmetic operators, shift operators, bitwise operators, sizeof operator	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	4	

2	Control structures	2.1 Conditional statements: Relational operators, logical operators, if statement, if-else statements, nested if-else statements, if-else ladder, switch statement. 2.2 Looping statements : 2.1 while loop, do… while loop, for loop. 2.3 Branching Statements: goto statement, use of 'break' and 'continue' statements.	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	8	1. Write a C program to find sum and average of three numbers. 2. Write a C program to find the sum of individual digits of a given positive integer. 3. Write a C program to generate the first n terms of the Fibonacci sequence. metrices from the console, verifies if metrics multiplication is possible or not. Then multiplies the metrices and prints the 3rd metrics. 4. a)Write a C program to generate prime numbers between 1 to n. 5. Write a C program to Check whether given number is Armstrong Number or Not. 6. Write a C program to evaluate algebraic expression (ax+b)/(ax-b). 7. Write a C program to check whether given number is perfect number or Not. 8. Write a C program to check whether given number is strong number or not. 9. Write a C program to check whether a number is Armstrongnumber or not. 10. Write a C program to check whether a number is Palindrome or not. 11. Write a C program to find maximum between two numbers. 12. Write a C program to find maximum between three numbers. 13. Write a C program to check whether a number is negative, positive or zero.

					14. Write a C program to check whether a number is divisible by 5 and 11 or not. 15. Write a C program to check whether a number is even or odd. 16. Write a C program to check whether a year is leap year or not. 17. Write a C program to check whether a character is alphabet or not. 18. Write a C program to input any alphabet and check whether it is vowel or consonant. 19. Write a C program to input any character and check whether it is alphabet, digit or special character.
3	Arrays and structure	3.1 Characteristics of an array, One dimension and two dimensional arrays, concept of multi-dimensional arrays. 3.2 Array declaration and Initialization. 3.3 Operations on Arrays. 3.4 Character and String input/output and String related operations. 3.5 Introduction and Features of Structures, Declaration and Initialization of Structures, array of structures. 3.6 Type def, Enumerated Data Type	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	8	1) Write a program to store marks for n number of student in an array and print their marks. 2) Write a program which stores the marks of subject Mathematics and English of n number of students in an array and then prints their individual total marks. 3) Write a program to insert an element in an array in a particular position. 4) Write a program to delete an element from a particular position of an array. 5) Write a program to convert a decimal number taken as input from user to corresponding binary number and store the result in an array. 6) Write a program to input a binary number in an array and convert into corresponding decimal number. 7) Write a program to find the smallest and the largest elements in an array.

					8) Write a program for deleting duplicate elements in an array. 9) Write a program to search a particular element in an array. 10) Write a program to sort n elements (ascending order). 11) Write a program to find second highest number from the array without using sorting. 12) Write a program to perform addition and subtraction between two matrices. 13) Write a program to transpose a matrix. 14) Write a program to add the elements of each row and each column of a matrix. 15) Write a program to perform multiplication of two matrices. 16) Write a program to check whether a matrix is identity matrix or not. 17) Write a program to check whether a matrix is sparse matrix or not 18) Write a C program to create a structure named company which has name, address, phone and no Of Employee as member variables. Read name of company, its address, phone and no Of Employee. Finally display these members‟ value. 19) Define a structure “complex” (typedef) to read two complex numbers and perform addition,subtraction of these two complex numbers and display the result.

					20) Write a C program to read RollNo, Name, Address, Age marks of 12 students in the BCT class and display the details from function.
4	Functions	4.1 Concept and need of functions. 4.2 Library functions: Math functions, String handling functions, other miscellaneous functions such as getchar(), putchar(), malloc(), calloc(). 4.3 Writing User defined functions - function definition, functions declaration, function call, scope of variables - local variables, global variables. 4.4 Function parameters: Parameter passing- call by value & call by reference, function return values, function return types ,declaring function return types, The 'return' statement. 4.5 Recursive functions.	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	7	1) Write a C program to add, subtract, multiply and divide two integers using userdefined type function with return type. 2) Write a C program to calculate sum of first 20 natural numbers using recursive function. 3) Write a C program to generate Fibonacci series using recursive function. 4) Write a C program to swap two integers using call by value and call by reference methods of passing arguments to a function. 5) Write a C program to find sum of digits of the number using Recursive Function. 6) Write a C program to read an integer number and print the reverse of that number using recursion. 7) Write a C program to find maximum and minimum between two numbers using functions. 8) Write a C program to check whether a number is even or odd using functions. 9) Write a C program to check whether a number is prime, Armstrong or perfect number using functions. 10) Write a C program to find power of any number using recursion.series using recursion.

5	Pointers	5.1 Introduction to Pointers : Definition, use of pointers, '*' and '&' operators, declaring, initializing, accessing pointers. 5.2 Pointer arithmetic. 5.3 Pointer to array. 5.4 Pointer and Text string. 5.5 Function handling using pointers. 5.6 Pointers to structure.	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	7	1. Write a C program to find the sum of all the elements of an array using pointers. 2. Write a C program to swap value of two variables using pointer. 3. Write a C program to add two numbers using pointers. 4. Write a C program to input and print array elements using pointer. 5. Write a C program to copy one array to another using pointer. 6. Write a C program to swap two arrays using pointers. 7. Write a C program to reverse an array using pointers. 8. Write a C program to search an element in array using pointers. 9. Write a C program to add two 2 X 2 matrix using pointers. 10. Write a C program to multiply two 2 X 2 matrix using pointers. 11. Write a C program to find length of string using pointers. 12. Write a C program to copy one string to another using pointer. 13. Write a C program to concatenate two strings using pointers. 14. Write a C program to compare two strings using pointers.

6		6.1Creation of the new file 6.2Opening an existing file 6.3Reading from the file 6.4Writing to the file 6.5Deleting the file			1. Write a C Program to list all files and sub-directories in a directory. 2. Write a C Program to count number of lines in a file. 3. Write a C Program to print contents of file. 4. Write a C Program to copy contents of one file to another file. 5. Write a C Program to merge contents of two files into a third file. 6. Write a C program to delete a file.

ListofBooksTextBooks:			
Nameof Author	**Title oftheBook**	**Edition/ISSN/ISBN**	**NameofthePublisher**
E.Balagurusamy	Programming in ANSI C	7thEd	McGraw Hill Education
ReferenceBooks: 2. ***Let us C*** by Yashavant Kanetkar, 19th Edition			

Subject Name:Data Structure with C **Credit: 4** **Lecture Hours:40 Subject Code: MCA103**

Name of the Course: **Data Structure with C**	
Course Code:MCA103 &MCA193	**Semester:1st**
Duration:40Hrs.	**Maximum Marks:100**
Teaching Scheme	**Examination Scheme**
Theory:3	End Semester Exam:100
Tutorial: 1	Continuous Assessment:100
Practical:2	Practical Sessional Internal continuous evaluation:100
Credit:4+2	Practical Sessional external examination:100

Aim:	
Sl.No.	
1	To gain Knowledge of Various aspects of algorithm development
2	To enhance the ability to identify the qualities of a good solution
3	To implement learned algorithm design techniques and data structures to solve problems.

Objective:	
Sl.No.	
1	The fundamental design,analysis and implementation of basic data structures.
2	Basic concepts in the specification and analysis of programs.

3	Principles for good program design,especially the uses of data abstraction.
4	Significance of algorithms in the computer field
Pre-Requisite:	
Sl.No.	
1.	Proficiency in one high level programming language
CourseOutcome:	
1.	On completion of this course students are expected to learn various data structures, their usages, merits andlimitations.
2.	On completion of this course students are expected to design and analyze various algorithms.
3.	On completion of this course students are expected to do a comparative analysis among different data structures and decide on the appropriate data structure to be used in a given scenario.
4.	On completion of this course students are expected to acquire adequate knowledge and skills to solve a real life software problem.
RelevantLinks:	
DSStudyMaterial	DS NPTELLINK DS CourseraLink DSLinkedInLearningLink

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO2	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-
CO3	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-
CO4	-	2	-	2	-	-	-	-	-	-	-	-	-	-	-

Module number	Topic	Sub-topics	MappingwithIndustry and International Academia	Lecture Hours	CorrespondingLabAssignment
1	**Algorithm Concept**	Algorithm concept, Time Complexity, Space Complexity, Running Time– Worst Case, Best Case, Average Case, time space trade-off, Algorithm Efficiency-Linear loops, Logarithmic loops, Nested loops, Time complexity comparison-Polynomial vs Exponential, Algorithm Notations-Big O , Big Omega, Theta Notation	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	4	

2	**Introduction to Data Structure, Array**	Program Efficiency, Data Structure-definition, usage, examples, Selection of Appropriate Data Structure, Data Structure-some terminologies, Classification of Data Structure, Fundamental difference between Linear and Non-linear Data Structure with examples, Operations on Linear Data Structure Introduction to Linear Data Structure-Array, 1D, 2D arrays, Row/Column major representation, sparse matrix	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	8	20. Write a C program to print an array. 21. Write a C program to check whether a given string is Palindrome or not. 22. Write a C program to convert temperature from degree Centigrade to Fahrenheit. 23. Write a C program to sort an array. 24. Write a C program to print the largest and second largest element of the array. 25. Write a C program to display Fibonacci series. 26. Write a program that reads two 2D metrices from the console, verifies if metrics multiplication is possible or not. Then multiplies the metrices and prints the 3rd metrics. 27. Write a program that reads a 2D metrics and checks if the metrics is a symmetric metrics or not. 28. Write a C program to print reverse array 29. Write a C program to check the sum of all elements of an array 30. Write a C program to check duplicate number in an array. 31. Write a C program to read a 2D array (with most of the elements as 0s) and then represent the same array as Sparse Metrics. 32. Write a C program to pass an array to a function using Call by Value, update the array values in the function, print the array elements both in the function and in the calling

					function. 33. Write a C program to pass an array to a function using Call by Reference, update the array values in the function, print the array elements both in the function and in the calling function. 34. Write a program to display n number of elements. Memory should be allocated dynamically using malloc( ). 35. Write a program to display n number of elements. Memory should be allocated dynamically using calloc( ). 36. Write a program to allocate memory using malloc( ) and then reallocate the previously allocated memory using realloc( ). Display the elements which have been taken after reallocation. 37. Write a program to allocate memory using calloc( ) and then reallocate the previously allocated memory using realloc( ). Display the elements which have been taken after reallocation. 38. Write a C program to search an element in an Array using dynamic memory allocation
3	**Linear Data Structure-Linked List**	Linked List-Introduction, Representation, Memory Allocation, Types- Singly, circular, doubly, doubly & circular, Operations on various linked lists-Count, Traverse/Display, Search, Insert, Delete	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed***	8	21) Write a Menu driven C program to accomplish the following functionalities in single linked list. a) Create a single linked list. b) Display the elements of a single linked list.

			syllabus: https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***IndustryMapping:*** The concepts delivered are in sync with the industry standards		c) Insert a node at the beginning of a single linked list. d) Insert a node at the end of a single linked list. e) Insert a node before a given node of a single linked list. f) Insert a node after a given node of a single linked list. g) Delete a node from the beginning of a single linked list. h) Delete a node from the end of a single linked list. i) Delete a node after a given node of a single linked list. j) Delete the entire single linked list. 22) Write a Menu driven C program to accomplish the following functionalities in circular linked list. a) Create a circular linked list. b) Display the elements of a circular linked list. c) Insert a node at the beginning of a circular linked list. d) Insert a node at the end of a circular linked list. e) Delete a node from the beginning of a circular linked list. f) Delete a node from the end of a circular linked list. g) Delete a node after a given node of a circular linked list. h) Delete the entire circular linked list. 23) Write a Menu driven C program to

					accomplish the following functionalities in doubly linked list. a) Create a doubly linked list. b) Display the elements of a doubly linked list. c) Insert a node at the beginning of a doubly linked list. d) Insert a node at the end of a doubly linked list. e) Insert a node before a given node of a doubly linked list. f) Insert a node after a given node of a doubly linked list. g) Delete a node from the beginning of a doubly linked list. h) Delete a node from the end of a doubly linked list. i) Delete a node after a given node of a doubly linked list. j) Delete the entire doubly linked list. 24) Write a Menu driven C program to accomplish the following functionalities in circular doubly linked list. a) Create a circular doubly linked list. b) Display the elements of a circular doubly linked list. c) Insert a node at the beginning of a circular doubly linked list. d) Insert a node at the end of a circular doubly linked list. e) Delete a node from the beginning of a circular doubly linked list.

					f) Delete a node from the end of a circular doubly linked list. g) Delete a node after a given node of a circular doubly linked list. h) Delete the entire circular doubly linked list.
4	**Linear Data Structure-Stack**	Introduction, Stack Operations – Push, Pop, Peek, Representation of Stack (Array, Linked List), Application of Stack: Reversing a list, Parentheses checker, Conversion of an infix expression into a postfix expression, Evaluation of a postfix expression, Conversion of an infix expression into a prefix Expression, Evaluation of a prefix expression, Recursion, Tower of Hanoi	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	7	11) Write a Menu driven C program to accomplish the following functionalities in Stack using an Array: a. Insert an element into the stack using an array (Push Operation). b. Delete an element from the stack using an array (Pop Operation). c. Return the value of the topmost element of the stack (without deleting it from the stack) using an array. d. Display the elements of a stack using an array. 12) Write a Menu driven C program to accomplish the following functionalities in Stack using Linked List: a. Insert an element into the stack using a Linked List (Push Operation). b. Delete an element from the stack using a Linked List (Pop Operation). c. Return the value of the topmost element of the stack (without deleting it from the stack) using a Linked List. d. Display the elements of the stack using a Linked List. 13) Write a program to convert an infix expression into its equivalent postfix notation. 14) Write a program to convert an infix

					expression into its equivalent prefix notation. 15) Write a program to evaluate a postfix expression. 16) Write a program to evaluate a prefix expression. 17) Write a program to print the Fibonacci series using recursion. 18) Write a program to solve the tower of Hanoi problem using recursion
5	**Linear Data Structure-Queue**	Introduction, Queue Operations – Enqueue, Dequeue, Peep, Representation of Queue (Array, Linked List), Types of Queues-Circular Queue, Deque, Priority Queue, Multiple Queue; Various operations (Enqueue, Dequeue, Peep) on the above mentioned queues-Both iterative & recursive implementation; Application of Queue	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***IndustryMapping:***The concepts delivered are in sync with the industry standards	7	1) Write a Menu driven C program to accomplish the following functionalities in Queue using an Array: a. Insert an element into the queue using an array (Enqueue Operation). b. Delete an element from the queue using an array (Dequeue Operation). c. Return the value of the FRONT element of the queue (without deleting it from the queue) using an array (Peep operation). d. Display the elements of a queue using an array. 2) Write a Menu driven C program to accomplish the following functionalities in Queue using Linked List: a. Insert an element into the queue using a Linked List (Enqueue Operation). b. Delete an element from the queue using a Linked List (Dequeue Operation). c. Return the value of the FRONT element of the queue (without deleting it from the queue) using a Linked List (Peep operation). d. Display the elements of a queue using a Linked List.

					3) Write a Menu driven C program to accomplish the following functionalities in Circular Queue using Array: a. Insert an element into the circular queue. b. Delete an element from the circular queue. c. Return the value of the FRONT element of the circular queue (without deleting it from the queue). d. Display the elements of a circular queue using the circular queue
6	**Searching & Sorting**	Searching- Types of Searching (Linear Search, Binary Search, Interpolation Search), Comparison among various Searching techniques Sorting-Types, Methods (Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort), Technique, Explanation, Algorithm and Examples on various sorting methods, Comparison of various sorting algorithms in terms of time complexity (Average case, Worst case)		6	1) Write a C program to implement the concept of Bubble sort. 2) Write a C program to implement the concept of Selection sort. 3) Write a C program to implement the concept of Insertion sort. 4) Write a C program to implement the concept of Quick sort. 5) Write a C program to implement the concept of Merge sort. 6) Write a C program to show that Quick sort is better than Bubble sort. 7) Write a C program to show that merge sort is more effective than quick sort. 8) Write a C program to search an element in an array using linear search. 9) Write a Cprogram to search an element in an array using binary search. 10) Write a C program to search an element in an array using interpolation search.

List of Books/Text Books:			
Name of Author	**Title of the Book**	**Edition/ISSN/ISBN**	**Name of the Publisher**
Reema Thareja	Data Structure Using C	2nd Ed	Oxford

Reference Books:			
Kruse, Tondo & Leung	Data Structures & Program Design in C	2nd Ed	PHI
Loudan	Mastering Algorithms With C		SPD/O'REILLY
Radhaganesan	C and Data Structures		Scitech Publications
Tenenbaum	DataStructureUsing C& C++	2nd Ed	PEI

Subject Name: Discrete Mathematical Structure **Credit: 3 Lecture Hours:41**

Subject Code: MCA104

Name of the Course: Discrete Mathematical Structure	
CourseCode: MCA104	**Semester:1**
Duration:41Hrs	**Maximum Marks:100**
Teaching Scheme	**Examination Scheme**
Theory:3	End Semester Exam:100
Tutorial: 1	Continuous Assessment:100
Credit: 3	
Aim:	
Sl.No.	
1	**Foundational Knowledge**: Establish a strong understanding of set theory, relations, propositional logic, permutations, combinations, generating functions, and recurrence relations.
2	**Graph Theory and Algorithms**: Develop proficiency in graph theory concepts, including sub-graphs, cyclic graphs, trees, spanning trees, and binary trees, along with mastering key algorithms such as Kruskal's, Prim's, Dijkstra's, Floyd's, Warshall's, DFS, and BFS.
3	**Automata and Formal Languages**: Gain expertise in constructing and converting NFA and DFA, state minimization, Mealy and Moore machines, understanding grammars (Types 0-3), and basic properties of fuzzy sets.
Objective:	
Sl.No.	

1	**Understand core Concepts**: Grasp fundamental concepts in set theory, relations, propositional logic, and combinatorics.
2	**Master Graph Theory**: Learn key graph theory principles, including graph types, trees, and essential graph algorithms.
3	**Explore Automata and Formal Languages**: Acquire skills in finite automata, state minimization, and the classification and use of formal grammars.
4	**Apply Advanced Topics**: Apply knowledge of recurrence relations, generating functions, and fuzzy sets to solve complex problems in mathematics and computer science.
Pre-Requisite:	
Sl.No.	
1.	Basic understanding of algebra and familiarity with mathematical reasoning and proof techniques.
CourseOutcome:	
1.	**Solid Foundation**: Develop a strong understanding of set theory, relations, and propositional logic.
2.	**Graph Theory Proficiency**: Gain expertise in graph theory concepts and algorithms, enabling efficient problem-solving.
3.	**Automata Skills**: Master the construction, conversion, and minimization of finite automata, and understand formal grammars.
4.	**Advanced Problem Solving**: Apply generating functions, recurrence relations, and fuzzy set theory to tackle complex mathematical and computational problems.
RelevantLinks:	
DS Study Material DS NPTELLINK https://onlinecourses.nptel.ac.in/noc20_cs82/preview DS CourseraLinkhttps://www.coursera.org/specializations/discrete-mathematics DSLinkedInLearningLink	

https://www.linkedin.com/learning/programming-foundations-discrete-mathematics/basics-of-discrete-mathematics

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO 1	3	2	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 2	2	3	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 3	3	3	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 4	3	3	-	2	2	-	-	-	-	-	-	-	3	-	-

Module number	Topic	Sub-topics	MappingwithIndustry and International Academia	Lecture Hours
1	**Set Theory and Functions**	**Set Theory:** 1. Definitions and operations (union, intersection, complement) 2. Power sets and Cartesian products. **Functions:** 1. Definitions and types (injective, surjective, bijective). 2. Composition of functions and inverse functions.	**Industry:**Data modeling, database management **Academia:**Foundations of computer science, mathematical analysis.	4

2	**Relations and Propositional Logic**	**Relations:** 1. Definitions and properties (reflexive, symmetric, transitive). 2. Equivalence relations and partitions. 3. Partial orders and Hasse diagrams. **Propositional Logic:** 1. Propositions and logical connectives. 2. Truth tables and logical equivalences. 3. Normal forms (CNF, DNF).	**Industry:**Software development, logic circuits, artificial intelligence. **Academia:**Formal methods, logic in computer science.	8
3	**Combinatorics and Mathematical Inductions**	**Combinatorics:** 1. Permutations and combinations. 2. Binomial theorem and Pascal's triangle. 3. Inclusion-exclusion principle. **Mathematical Inductions** 1. Principle of mathematical induction. 2. Strong induction. 3. Applications and examples.	**Industry:**Cryptography, algorithm design, network security. **Academia:**Discrete mathematics, theoretical computer science.	8

4	**Graph Theory and Algorithms**	**Graph Theory** 1. Definitions and types of graphs. 2. Sub-graphs, cyclic graphs, and trees. 3. Spanning trees and binary trees. **Graph Algorithms** 1. Kruskal's and Prim's algorithms (minimum spanning trees). 2. Dijkstra's algorithm (shortest path). 3. Floyd-Warshall algorithm (all-pairs shortest paths). 4. DFS and BFS (graph traversal).	**Industry:**Network analysis, operations research, data science. **Academia:**Algorithm design, computational complexity.	8
5	**Automata and Formal Languages**	**Automata** 1. Definitions and differences between NFA and DFA. 2. Conversion of NFA to DFA. 3. State minimization techniques. 4. Mealy and Moore machines. **Formal Languages** 1. Grammar types (Type 0, 1, 2, 3). 2. Chomsky hierarchy.	**Industry:**Compiler design, text processing, machine learning. **Academia:**Automata theory, formal language theory.	7

		3. Regular expressions and languages.		
6	**Advanced Topics**	**Generating Functions** 1. Definitions and basic properties. 2. Applications in counting and solving recurrences. **Recurrence Relations** 1. Linear recurrence relations with constant coefficients. 2. Methods of solving recurrences (characteristic equation, generating functions). **Fuzzy Sets** 1. Definitions and basic properties. 2. Operations on fuzzy sets (union, intersection, complement). 3. Applications and examples.	**Industry:**Financial modeling, operations research. **Academia:**Discrete mathematics, combinatorial analysis, Soft computing, artificial intelligence.	6

ListofBooksTextBooks:			
Nameof Author	**Title oftheBook**	**Edition/ISSN/ISBN**	**NameofthePublisher**
Rosen	Discrete Mathematics With Applications	7th	McGraw Hill
ReferenceBooks:			
Somasundaram	Discrete Mathematical Structure		PHI

Dubey	Discrete Mathematical Structure		EXCEL BOOKS
Bhisma Rao	Discrete Structure and Graph Theory		Scitech
G.S.RAO	Discrete Mathematical Structure		New Age International
Mott, Kandel & Baker	Discrete Mathematics for Comp. Scientists & Mathematician		PHI

Subject Name:Business English and Communication

Credit: 3 **Lecture Hours:3** **Subject Code: MCA105**

Name of the Course: Business English and Communication	
Course Code:MCA105	**Semester: 1**
Duration:	**Maximum Marks:100**
Teaching Scheme	**Examination Scheme**
Theory: 3	End Semester Exam:100
Tutorial: 1	Continuous Assessment:100
Credit: 3	
Aim:	
Sl.No.	
1	Making the students industry-ready.
2	Making the students relevant in the contemporary society.
3	Making the students prepared to analyze and solve problems through listening, speaking, reading and writing skills.
Objective:	
Sl.No.	
1	To develop effective business writing and communication skills.
2	To enhance oral communication and presentation abilities among students.
3	To help students learn to prepare various business documents and technical reports.

4	To improve listening and reading comprehension.
Pre-Requisite:	
Sl.No.	
1.	Basic English Proficiency, Listening and Speaking Skills, Reading and Writing Skills, Academic and Social Contexts, and Familiarity withCorporateEthics.
CourseOutcome:	
1.	Achieve competence in grammar,syntax and vocabulary fundamentals.
2.	Effectively communicate in academic and social contexts.
3.	Developreadinessfor theindustryandunderstandcorporateethics.
4.	Acquire basic proficiency in English encompassing reading, listening, comprehension, writing, and speaking skills.
RelevantLinks:	
Study MateriaNPTEL	CourseraLinkedin Learning

COs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**
CO 1	3	2	-	2	-	2	-	1	2	3	-	3	3	-	-
CO 2	3	3	-	2	-	3	1	2	3	3	-	3	3	-	-

CO 3	2	3	-	2	-	3	1	3	2	3	-	3	3	-	-
CO 4	3	3	-	2	-	2	-	2	2	3	-	3	3	-	-

Module number	Topic	Sub-topics	MappingwithIndustry and International Academia	Lecture Hours
1	**Introduction to Business Communication.**	- Importance of effective communication in business. - Types of business communication: Internal and External. - Communication process and barriers. - Strategies for effective communication. - Traditional and digital communication channels. - Effective use of email, memos, and business letters. - Communication through social media and professional networks.	International academia: https://www.coursera.org/learn/understanding-corporate-communications	4

2	**Writing Skills Development**	- Formats and styles of business letters. - Writing formal and informal business letters. - Common types of business letters: Inquiry, Complaint, Application, Appreciation. - Structure of technical reports. - Writing abstracts, executive summaries, and conclusions. - Incorporating visuals and data in reports.	*InternationalAcademia:* https://ocw.mit.edu/courses/21g-222-expository-writing-for-bilingual-students-fall-2002/	8
3	**Oral Communication Skills**	- Preparing and delivering business presentations. - Using multimedia in presentations. - Techniques for effective public speaking. - Prepared speech exercises. - Extempore speech practice. - Role-playing business scenarios.	International Academia: https://ocw.mit.edu/courses/21g-222-expository-writing-for-bilingual-students-fall-2002/ Stanford Courses Online: https://online.stanford.edu/courses/csp-xcom88-high-impact-communication-advance-your-technology-career https://online.stanford.edu/courses/gsb-x0011-sharpen-your-communication-skills *IndustryMapping:* • CampusInterviews and recruitment drives. Software: Orell Talk	8

			https://orelltalk.com/	
4	**Listening and Reading Skills**	- Importance of active listening in business. - Techniques for improving listening skills. - Listening comprehension exercises. - Developing reading comprehension. - Strategies for effective reading. - Comprehension tests and exercises.	International Academia: https://ocw.mit.edu/courses/21g-222-expository-writing-for-bilingual-students-fall-2002/ Stanford Courses Online: https://online.stanford.edu/courses/csp-xcom88-high-impact-communication-advance-your-technology-career https://online.stanford.edu/courses/gsb-x0011-sharpen-your-communication-skills *IndustryMapping:* ● CampusInterviews and recruitment drives. ● Software: Orell Talk https://orelltalk.com/	
5	**Practical Communication Applications**	- Principles of organizing written material. - Structuring content for clarity and impact. - Editing and proofreading techniques - Designing effective posters for business presentations.	*InternationalAcademia*: https://ocw.mit.edu/courses/21g-222-expository-writing-for-bilingual-students-	7

		- Visual and textual balance. - Presenting posters in professional settings.	fall-2002/ *IndustryMapping:* • Emailwritingand writing other relevantcorporate documents. Software: Orell Talk https://orelltalk.com/	
6	**Practical Communication Skill Development**	- Interactive sessions on negotiation and persuasion. - Group discussions and teamwork exercises	International academia: ***https://ocw.mit.edu/courses/15-280-communication-for-managers-fall-2016/*** *IndustryMapping:* • CampusInterviews and recruitment drives.	6

List of Books/Text Books:			
Name of Author	**Title of the Book**	**Edition/ISSN/ISBN**	**Name of the Publisher**
R C Sharma and Krishna Mohan	BusinessCorrespondence &ReportWriting	ISBN 978-9385965050 (5th ed)	McGraw Hill Education
Reference Books:			
MatthukuttyMonippally	BusinessCommunicationStrategies	ISBN 978-0070435773	McGraw Hill Education
K.R. Lakshminarayanan	EnglishforTechnicalcommunication	Volume 1 & 2 Combined Edition	SCITECH PUBLICATIONS (INDIA) PVT LTD

Asha Kaul	Business Communication	Second Edition	PHI Learning
Dr. Anjali Ghanekar	Communication Skills for Effective Management	ISBN 978-8186314500 (19^{th} ed)	Everest Publishing House