DEPARTMENT OF COMPUTER APPLICATIONS

MCA STRUCTURED SYLLABUS - FIRST YEAR AND FIRST SEMESTER OF MCA2026 ADMISSION BATCH

Course Code	Course Title	Total No. of Weekly Contact Hours				Total No. of Credits
		Lecture (L)	Tutorial (T)	Practical (P)	Total Lectures	
1st Semester (Theory)						
MCA101	Computer Organisation and Architecture	4	1	0	5	3
MCA102	Computer Programming with C	4	1	0	5	4
MCA103	Data Structures with C	4	1	0	5	4
MCA105	Business English and Communication	3	1	0	4	3
MCA106	Computational Mathematics and Statistics	4	1	0	5	4
MCAGS101	Mental Maths for Professionals - I	2	1	0	3	1
Total of Theory					**27**	**19**
1st Semester (Practical)						
MCA192	C Programming Laboratory	0	0	4	4	3
MCA193	Data Structures Laboratory using C	0	0	3	3	3
Total of Practical					**7**	**6**
1st Semester (Sessional)						
MCA171	Research Methodology and IPR	2	1	0	3	2
MCA172	Python Programming Fundamentals	0	0	3	3	2
IFC	Industry and Foreign Certification	0	0	0	0	0
MAR	Mandatory Additional Requirements	0	0	0	0	0
MOOCs	Massive Open Online Course	0	0	0	0	0
Total of Sessional					**6**	**4**
Total of the Semester					**40**	**29**

Institute of Engineering & Management, Kolkata
Newtown
(School of University of Engineering and Management, Kolkata)

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

Subject Code: MCA101 **Lecture Hours: 40**

Course Name: Computer Organisation and Architecture	
Course Code: MCA101	**Semester: 1st**
Duration: 40 Hrs.	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 4	End Semester Exam: 100
Tutorial: 1	Continuous Assessment: 100
Credit: 4	

Aim:	
Sl. No.	
1	To have a thorough understanding of the basic structure and operation of a digital computer.
2	To study the different communication methods 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 of Pipeline processing, RISC and CISC architectures.
4	Develop a base for advanced microprocessors.
Pre-Requisite:	
Sl. No.	
1.	Proficiency in basic Digital Electronics
Course Outcome:	
1.	Summarize the fundamental components of a basic computer system and its organization.
2.	Apply arithmetic and logical microoperations of binary number systems.
3.	Analyze control unit design and concept of pipelining.
4.	Classify memory hierarchy and examine numerical problems based on it.

CO-PO-PSO Mapping

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	Mapping with Industry and International Academia	Lecture Hours
1	**Structure of Computers and Computer Arithmetic**	Computer types, Functional units, Basic operational concepts, von Neumann 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:*** https://www.aicte-india.org/downloads/mcadegree.pdf ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	4
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 ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	8
3	**Register Transfer, 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 ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	8
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://www.aicte-india.org/downloads/mcadegree.pdf ***Industry Mapping:*** 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 ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	7
6	**MULTI-PROCESSORS**	Characteristics of multiprocessors, Interconnection structures, Inter Processor Arbitration, Interprocessor Communication and Synchronization, and Cache Coherence.	***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 ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	6

List of Books Text Books:			
Name of Author	**Title of the Book**	**Edition/ISSN/ISBN**	**Name of the Publisher**
M. Moris Mano	Computer System Architecture	3rd Ed	Pearson/PHI
Reference Books: **1. Carl Hamacher, Zvonks Vranesic, Safea Zaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India.**			

Institute of Engineering & Management, Kolkata
Newtown
(School of University of Engineering and Management, Kolkata)

Subject Name: Computer Programming with C **Credit: 4**

Subject Code: MCA102 **Lecture Hours: 48**

Course Name: Computer Programming with C & C Programming Laboratory	
Course Code: MCA102 & MCA192	**Semester: 1st**
Duration: 40 Hrs.	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 4	End Semester Exam: 100
Tutorial: 1	Continuous Assessment: 100
Practical: 4	Practical Sessional Internal continuous evaluation: 100
Credit: 4+3	Practical Sessional external examination: 100

Aim:	
Sl. No.	
1	To develop a rigorous understanding of algorithmic problem-solving and translate abstract logic into structured C programs.
2	To cultivate the ability to write modular, efficient, and memory-safe code suited for postgraduate-level software development.
3	To establish a strong foundation in core programming constructs, memory management, and data handling as a stepping stone for advanced systems programming.

Objective:	
Sl. No.	
1	A foundational understanding of computer operations, mathematical logic, and an aptitude for computational problem-solving.
2	To impart an in-depth understanding of memory management, specifically through pointer arithmetic and dynamic memory allocation.

3	To demonstrate the practical implementation of user-defined data types (structures and unions) and secure string manipulations.
4	To foster critical computational thinking necessary for optimising code, debugging errors, and implementing file handling for data persistence.
Pre-Requisite:	
Sl. No.	
1.	Proficiency in one high-level programming language
Course Outcome:	
Upon successful completion of this course, students will be able to:	
1.	Apply algorithmic thinking and fundamental C programming constructs (control structures, operators) to translate mathematical and logical problems into executable code.
2.	Analyse computational problems to select and implement appropriate linear data structures (arrays, strings) for efficient data organisation and manipulation.
3.	Evaluate memory management and program modularity by leveraging pointers, dynamic memory allocation, and user-defined functions to optimise software performance.
4.	Design and develop comprehensive software solutions utilizing heterogeneous data types (structures/unions) and persistent file I/O operations for real-world data processing scenarios.

CO-PO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	1	2	1	-	1	1	1	1	2	3	1	2
CO2	2	3	3	2	2	1	-	1	1	1	1	2	3	2	2
CO3	2	2	3	2	3	1	1	1	1	1	2	2	3	2	2
CO4	2	2	3	2	3	2	1	1	2	1	2	2	3	3	2

Module Number	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
1	**Fundamentals & C Basics**	Algorithmic Thinking & Compilation Process Tokens, Data Types, and Memory Footprints Formatted & Unformatted I/O Operators, Expressions, and Type Casting	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standard	**6**	Week 1: Fundamentals, I/O, and Operators
2	**Control Structures**	if, else, and switch statements for, while, and do-while loops Nested loops and optimization Jump statements (break, continue, goto)	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	**6**	Week 2: Control Structures
3	**Functions & Storage Classes**	Modular design, Declaration, and Call Stack Parameter passing and Return values Recursion mechanics and examples Storage Classes and Scope rules	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	**7**	Week 3: Functions & Recursion
4	**Arrays and Strings**	1D Arrays and passing arrays to functions 2D Arrays and Matrix operations	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50**	**7**	Week 4: 1-D and 2-D Arrays Week 5: Strings

		String basics and <string.h> library Secure coding (preventing buffer overflows)	**(Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards		& Secure Coding
5	**Pointers & Dynamic Memory Allocation**	Pointer fundamentals and Address mechanics Pointer arithmetic and advanced types (void, null) Pointers with Arrays and Strings Pass by Reference Dynamic Memory (malloc, calloc, free, leaks)	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	**9**	Week 6: Pointers Fundamentals Week 7: Dynamic Memory Allocation (DMA)
6	**User-Defined Data Types**	Structures (Basics, Array of Structures, Nested) Passing Structures and the -> operator Memory alignment and padding Unions, enum, and typedef	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	**5**	Week 8: User-Defined Data Types
7	**File Handling & Advanced Concepts**	File operations (Open, Read, Write, Close modes) Random Access (fseek, ftell, rewind) Command Line Arguments Preprocessor Directives	***International Academia:*** • **Massachusetts Institute of Technology (MIT): 6.087 (Practical Programming in C)** • **Harvard University: CS50 (Introduction to Computer Science)** ***AICTE-prescribed syllabus (3 Years):*** ***AICTE-prescribed syllabus (2 Years):*** ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	**5**	Week 9: File Handling & Command Line Arguments Weeks 10 & 11: Capstone Projects

List of Books Text Books:			
Name of Author	**Title of the Book**	**Edition/ISSN/ISBN**	**Name of the Publisher**
E. Balagurusamy	Programming in ANSI C	978-9389949186	McGraw-Hill Education
Yashavant Kanetkar	Let us C	978-9365899689	BPB Publications
K R Venugopal; Sudeep R Prasad	Mastering C	978-9332901278	McGraw-Hill Education
Reference Books:			
Dennis Ritchie; Brian W. Kernighan	C PROGRAMMING LANGUAGE, 2ND EDN	978-9332549449	**Pearson Education India**
K. N. King	C Programming, A Modern Approach, 2nd Edition	978-0-393-97950-3	**W. W. Norton & Company**

Annexure – I

Course Code: MCA102

Course Name: C Programming Lab

Week 1: Fundamentals, I/O, and Operators

Direct Questions:

1. Write a program using the `sizeof()` operator to print the memory footprint of `char`, `int`, `float`, and `double` on your current compiler architecture.
2. Write a program to swap the values of two variables without using a third temporary variable.
3. Write a program that takes an integer input from the user and uses bitwise operators to determine if the number is odd or even.

Story-Based Question: *A new music streaming startup needs a basic bandwidth calculator to estimate server load. Write a program that asks the user for the number of active listeners, the average audio bitrate (in kbps), and the average song duration (in minutes). Calculate and display the total data transferred in Megabytes (MB).*

Week 2: Control Structures

Direct Questions:

1. Write a menu-driven program using a `switch-case` statement to simulate a basic calculator. Handle the divide-by-zero edge case.
2. Write a program to print a right-angled pyramid pattern of numbers.
3. Write a program to check if a user-inputted number is an Armstrong number.

Story-Based Question: *A local research team is analyzing epidemiological data to track a recent outbreak. Write a program that takes a patient's data as input: Age, Body Temperature, and Symptom Duration. Use an* `else-if` *ladder to classify their risk level as "Low," "Moderate," or "High" based on threshold conditions you define.*

Week 3: Functions & Recursion

Direct Questions:

1. Write a function `isPrime(int n)` that returns 1 if prime and 0 otherwise. Print all prime numbers between 1 and 100.
2. Write a recursive function to calculate the Factorial of a given number.
3. Write a user-defined function `power(int base, int exp)` that calculates base exponentiation without using `<math.h>`.

Story-Based Question: *For an upcoming university CyberX workshop, participants need a secure, dynamically generated passcode. Write a recursive function that takes a 4-digit ID and recursively sums the digits until a single digit is obtained. Multiply that digit by a security constant to generate the passcode.*

Week 4: 1-D and 2-D Arrays

Direct Questions:

1. Write a program to find the second largest and second smallest elements in a 1D array of integers.
2. Write a program to perform Matrix Multiplication on two 3x3 matrices.
3. Write a program to reverse the elements of a 1D array *in-place*.

Story-Based Question: *You are organizing an online Capture the Flag (CTF) programming competition. You have two separate, sequentially sorted arrays of competitor scores. Write a program to seamlessly interleave and merge these two arrays into a single sorted array to generate the final leaderboard.*

Week 5: Strings & Secure Coding

Direct Questions:

1. Write a program to find the length of a string and reverse it without using standard library functions.
2. Write a program that takes a string input and counts the total number of vowels, consonants, digits, and whitespaces.
3. Write a program to check if two strings inputted by the user are anagrams.

Story-Based Question: *You are formatting an academic research paper on XAI, and the publisher requires specific proprietary frameworks to be redacted. Write a program that reads a text using* `fgets()`*, searches for a target word (like "SHAP" or "LIME"), and replaces all its occurrences with asterisks (****).*

Week 6: Pointers Fundamentals

Direct Questions:

1. Write a program to swap two variables using pass-by-reference.
2. Traverse a 1D integer array and print its elements using *only* pointer arithmetic.
3. Write a function that returns both the maximum and minimum values of an array via pointer parameters.

Story-Based Question: *A smart wearable device tracks* `heart_rate` *and* `step_count`*. Write a function taking these variables by reference. If the* `heart_rate` *exceeds 150 BPM, reset the* `step_count` *to zero and double the* `heart_rate` *to trigger a hardware alert state.*

Week 7: Dynamic Memory Allocation (DMA)

Direct Questions:

1. Use `malloc()` to allocate memory for an array of N integers. Populate, calculate the sum, and `free()` the memory.
2. Demonstrate `calloc()` by allocating memory for an array of floats and verifying initialization to zero.
3. Use `realloc()` to expand a dynamically allocated array of 5 integers to hold 10 integers.

Story-Based Question: *An epidemiological prognosis system needs to load daily patient records. Write a program that asks for the current day's patient count, dynamically allocates the memory to store their ages, calculates the moving average age, and strictly ensures the memory is freed before termination.*

Week 8: User-Defined Data Types

Direct Questions:

1. Define a `struct Employee` (emp_id, name, salary). Read and print data for one employee.
2. Create an array of `struct Student`. Read data for 5 students and print the details of the one with the highest marks.
3. Write a program demonstrating a `union`, showing how memory is shared and overwritten.

Story-Based Question: *An online music learning platform manages its catalog using structures. Define a* `Course` *structure containing an inner* `Pricing` *structure. Create an array of 3 courses, calculate the final price after discounts for each, and display the course with the lowest final cost.*

Week 9: File Handling & Command Line Arguments

Direct Questions:

1. Write a program using command-line arguments to copy the contents of one file into another.
2. Write a program to open a text file and count the number of lines, words, and characters.
3. Accept an integer via command-line and append its multiplication table to `table.txt`.

Story-Based Question: *A cybersecurity digital footprint analyzer monitors network requests. Write a program that reads a mock log file. Extract any line containing the keyword "UNAUTHORIZED" and append it to a new file called* `threat_report.txt`*.*

Weeks 10 & 11: Capstone Projects

*Students must select **ONE** of the following 12 capstone projects to complete. These projects require the integration of multiple core concepts including Pointers, Structures, Dynamic Memory, and File Handling.*

Project 1: The Epidemiological Prognosis Tracker

An epidemiological research lab needs a system to process daily patient metrics. Write a C program that reads a dataset of patients (Patient ID, Age, Body Temperature, Symptom Duration) from a file. Using structures and pointer manipulation, apply a rules-based threshold to identify high-risk cases. The program must automatically extract these flagged records and generate a formatted "Prognosis_Report.txt" for the clinical team.

Project 2: Music Startup Ecosystem Manager

A new startup focusing on musical education and talent discovery needs a backend inventory engine. The platform manages three distinct entities: "Learning Courses", "Instrument Access", and "Talent Profiles". Using arrays of structures and dynamic memory allocation (`malloc`/`realloc`), build a menu-driven program that allows administrators to add new courses, update instrument availability, and display a sorted list of registered musical talents.

Project 3: CyberX Digital Footprint Analyzer

Following the successful conclusion of a CyberX cybersecurity workshop, the server generated a massive log file of network access requests. Write a command-line application that parses this "digital footprint" text file line by line. Using advanced string manipulation, the program must extract unique IP addresses, count their frequency, and quarantine any log entry containing the word "UNAUTHORIZED" into a separate secure text file.

Project 4: Capture The Flag (CTF) Auto-Scorer

You are tasked with building the backend scoring engine for an upcoming university Hackerrank-style CTF competition. Competitor data (Team Name, Challenges Solved, Time Penalties) is continuously updated. Use pointers to structures to manage the team data. Write a dynamic sorting algorithm that instantly updates the leaderboard based on net scores and displays the top 3 teams in real-time on the console.

Project 5: Explainable AI (XAI) Feature Importance Matrix

In machine learning, algorithms like SHAP and LIME output feature importance scores. Write a C program that simulates this by generating a 2D matrix of floating-point numbers representing "feature weights" for a dataset. Calculate the row-wise and column-wise averages. Apply a user-defined threshold to identify the most critical data features, extracting their indices into a 1D dynamically allocated array using pointers.

Project 6: University Event Poster Layout Generator

The Department of Computer Applications frequently hosts workshops and competitions. Write a text-based layout generator that takes user input for an event (Title, Date, Time, Venue, Speaker Name). Using string formatting, loop constructs, and standard I/O padding, the program must generate a visually appealing ASCII-art "poster" layout and save it as a text file to be distributed to the MCA students.

Project 7: Supermarket Inventory & Billing System

Build a robust point-of-sale system for a local supermarket. Define a structure for `Product` (ID, Name, Price, Stock). The program should allow a cashier to input purchased item IDs and quantities. It must check stock availability, calculate the total bill including a 5% GST, update the remaining inventory using pass-by-reference, and print an itemized receipt to the console.

Project 8: Advanced Cryptographic Secure Messaging

Design a two-way secure messaging utility. The program takes a string message from the user and a numeric encryption key. Implement a custom encryption algorithm that applies a Caesar Cipher shift followed by a bitwise XOR operation on every character. The program must be able to output the encrypted ciphertext, and conversely, take ciphertext and the key to decrypt it back to the original message.

Project 9: The Hospital Blood Bank Matcher

A regional hospital requires a fast blood-matching registry. Store donor records (Name, Blood Group, Age, Contact) in a binary file using `fwrite()`. When an emergency trauma request comes in, the user inputs the required blood type and units needed. The program must rapidly search the binary file using `fread()`, identify eligible donors, and output their contact information, ensuring all file pointers are securely closed.

Project 10: Dynamic Library Book Allocator

A university library's new acquisitions arrive in unpredictable batch sizes. Write a program that initially allocates memory for an array of 5 `Book` structures (Title, Author, ISBN). As the user inputs new books, if the array becomes full, use `realloc()` to dynamically double the array size without losing previous entries. Implement a function to safely `free()` the entire catalog when the program terminates.

Project 11: Railway Route & Seat Manager

Design a train ticketing system for a specific route. Define a structure for passenger tickets. Manage a 1D array representing 50 available seats. The menu-driven application must allow users to view available seats, book a specific seat number, cancel a ticket, and display the final passenger manifest. Prevent double-booking using logical condition checks.

Project 12: MCA Gradebook & SGPA Calculator

Create a comprehensive gradebook application for the 1st-semester MCA cohort. The program reads a file containing student Roll Numbers and marks in 5 subjects. Using user-defined functions, calculate the total marks, percentage, and SGPA for each student based on the university's grading scale. The application must generate individual text-file "report cards" named dynamically (e.g., "Report_Roll101.txt").

University of Engineering and Management
Institute of Engineering & Management, New Town Campus

Subject Name: Data Structure with C **Credit: 4**

Code: MCA103 **Lecture Hours: 40**

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

Aim:	
Sl. No.	
1	To gain Knowledge of Various aspects of algorithm development
2	To enhance Ability to identify 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
Course Outcome:	
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 structuresand 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 lifesoftware problem.

CO-PO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	-	-	-	-	-	-	1	1	2	2
CO2	3	3	1	2	1	-	-	-	-	-	-	2	3	2	1
CO3	3	2	2	3	2	-	-	-	-	-	-	1	3	2	1
CO4	2	2	3	2	2	-	-	-	-	-	-	2	3	2	2

Module number	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignment
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 ***Industry Mapping:*** 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 ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	8	1. Write a C program to print an array. 2. Write a C program to check whether a given string is Palindrome or not. 3. Write a C program to convert temperature from degree Centigrade to Fahrenheit. 4. Write a C program to sort an array. 5. Write a C program to print the largest and second largest element of the array. 6. Write a C program to display Fibonacci series. 7. 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. 8. Write a program that reads a 2D metrics and checks if the metrics is a symmetric metrics or not. 9. Write a C program to print reverse array 10. Write a C program to check the sum of all elements of an array

					11. Write a C program to check duplicate number in an array. 12. 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. 13. 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. 14. 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. 15. Write a program to display n number of elements. Memory should be allocated dynamically using malloc( ). 16. Write a program to display n number of elements. Memory should be allocated dynamically using calloc( ). 17. 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. 18. 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. 19. 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 syllabus:*** https://makautexam.net/aict	8	1. 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. c) Insert a node at the beginning of a single linked list.

			e_details/Syllabus/MCA/sem221.pdf ***Industry Mapping:*** The concepts delivered are in sync with the industry standards		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. 2. 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. 3. 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. 4. 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 ***Industry Mapping:*** The concepts delivered are in	7	1. 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. 2. Write a Menu driven C program to accomplish the following functionalities in Stack using

			sync with the industry standards		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. 3. Write a program to convert an infix expression into its equivalent postfix notation. 4. Write a program to convert an infix expression into its equivalent prefix notation. 5. Write a program to evaluate a postfix expression. 6. Write a program to evaluate a prefix expression. 7. Write a program to print the Fibonacci series using recursion. 8. 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 ***Industry Mapping:*** 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). 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)	***International Academia:*** https://web.stanford.edu/class/cs97si/03-data-structures.pdf ***AICTE-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/MCA/sem221.pdf ***Industry Mapping:*** The concepts delivered are in sync with the industry standards	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 C program 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:			
Tenenbaum	Data Structure Using C & C++	2nd Ed	PEI
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

University of Engineering and Management
Institute of Engineering & Management, New Town Campus

Subject Name: Business English and Communication **Credit: 3**

Subject Code: MCA105 **Lecture Hours: 33**

Name of the Course: Business English and Communication	
Course Code: MCA105	**Semester: 1**
Duration: 33	**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 with Corporate Ethics.
Course Outcome:	
1.	Achieve competence in grammar, syntax, and vocabulary fundamentals.
2.	Effectively communicate in academic and social contexts.
3.	Develop readiness for the industry and understand corporate ethics.
4.	Acquire basic proficiency in English encompassing reading, listening, comprehension, writing, and speaking skills.

CO-PO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	2	-	2	-	1	2	3	-	3	3	-	-
CO2	3	3	-	2	-	3	1	2	3	3	-	3	3	-	-
CO3	2	3	-	2	-	3	1	3	2	3	-	3	3	-	-
CO4	3	3	-	2	-	2	-	2	2	3	-	3	3	-	-

Module number	Topic	Sub-topics	Mapping with Industry 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, and Appreciation. - Structure of technical reports. - Writing abstracts, executive summaries, and conclusions. - Incorporating visuals and data in reports.	International Academia: 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 *Industry Mapping:* Campus Interviews and Recruitment Drives. Software: Orell Talk https://orelltalk.co m/	8

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 *Industry Mapping:* • Campus Interviews and recruitment drives. • Software: Orell Talk https://orelltalk.co m/	
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. - Visual and textual balance. - Presenting posters in professional settings.	*International Academia*: https://ocw.mit.edu/courses /21g-222-expository- writing-for-bilingual-students-fall-2002/ *Industry Mapping:* • Email writing and writing other relevant corporate documents. Software: Orell Talk https://orelltalk.co m/	7

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/* *Industry Mapping:* • Campus Interviews 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	Business Correspondence & Report Writing	ISBN 978-9385965050 (5th ed)	McGraw Hill Education
Reference Books:			
Matthukutty Monippally	Business Communication Strategies	ISBN 978-0070435773	McGraw Hill Education
K.R. Lakshminarayanan	English for Technical Communication	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 (19th ed)	Everest Publishing House

University of Engineering and Management
Institute of Engineering & Management, New Town Campus

Subject Name: Computational Mathematics and Statistics **Credit: 4**

Lecture Hours: 48 **Subject Code: MCA106**

Course Name: Computational Mathematics and Statistics	
Course Code: MCA106	**Semester: 1st**
Duration: 48 Hrs.	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 4	End Semester Exam: 100
Tutorial: 1	Continuous Assessment: 100
Credit: 4	

Aim:	
Sl. No.	
1	To provide a rigorous foundation in discrete mathematical structures, including set theory and propositional logic, which are essential for software development and formal methods.
2	To develop analytical and statistical problem-solving skills necessary for data-driven computing, focusing on probability distributions, parameter estimation, and statistical hypothesis testing.
3	To introduce advanced mathematical concepts like fuzzy sets and generating functions that underpin modern computational fields such as artificial intelligence, soft computing, and operations research.

Objective:	
Sl. No.	
1	Introduce the fundamental concepts of relations, functions, and propositional logic (including CNF and DNF) to formulate formal computational models and logic circuits.
2	Equip students with combinatorial tools, including the inclusion-exclusion principle, and mathematical induction techniques necessary for analyzing algorithm design and cryptography.
3	Teach foundational statistical measures, probability distributions (Binomial, Poisson, Normal), and hypothesis testing methods (z-test, t-test, chi-squared test, ANOVA) required for machine learning and numerical analysis.
4	Explore advanced mathematical frameworks, specifically generating functions for solving recurrences and fuzzy set operations for applications in financial modeling and soft computing.
Pre-Requisite:	
Sl. No.	
1.	A fundamental understanding of high-school-level algebra and introductory algorithmic thinking to effectively grasp abstract logic, recurrences, and numerical computations.
Course Outcome:	
Upon successful completion of this course, students will be able to:	
1.	Apply the principles of set theory, mathematical relations, and propositional logic to solve foundational discrete computational problems and model logic circuits.
2.	Analyze combinatorial problems and algorithm constraints using permutations, combinations, and the principle of mathematical induction.
3.	Evaluate datasets and statistical models by selecting and applying appropriate probability distributions and hypothesis testing methods (large and small sample tests) to assess data consistency and error.
4.	Formulate computational solutions for complex systems using advanced mathematical models, such as generating functions for recurrences and fuzzy sets for artificial intelligence applications.

CO-PO-PSO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	-	1	2	-	1
CO2	3	3	2	1	-	-	-	-	-	-	-	1	2	-	1
CO3	3	3	1	3	2	-	-	-	-	-	-	1	1	3	2
CO4	3	3	2	2	1	-	-	-	-	-	-	2	1	2	3

Module number	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours
1	**Set Theory and Functions**	Set Theory: Definitions and operations (union, intersection, complement), Power sets and Cartesian products. Functions: Definitions and types (injective, surjective, bijective), Composition of functions and inverse functions.	Functions: Definitions and types (injective, surjective, bijective), Composition of functions and inverse functions.	4
2	**Relations and Propositional Logic**	Relations: Definitions and properties (reflexive, symmetric, transitive), Equivalence relations and partitions, Partial orders and Hasse diagrams. Propositional Logic: Propositions and logical connectives, Truth tables and logical equivalences, 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: Permutations and combinations, Binomial theorem and Pascal's triangle, Inclusion-exclusion principle. Mathematical Inductions: Principle of mathematical induction,	Industry: Cryptography, algorithm design, network security. Academia: Discrete mathematics, theoretical computer science	8
4	**Probability Distributions & Statistics**	Statistics - measure of central tendency, dispersion (Moments, Skewness & Kurtosis). Least square curve fitting Introduction to mass function, density function, distribution function (Binomial, Poisson, Normal), estimation of parameters (unbiasedness-concept of noise/error, consistency). Testing of Hypothesis, Large Sample, Small Sample (z test, t test, chi-squared test, Anova)	**Industry Mapping:** https://www.sagemath.org/ , MATLAB **International Academia:** https://ocw.mit.edu/courses/18-440-probability-and-random-variables-spring-2014/pages/lecture-notes/ ,	6 6

			https://ocw.mit.edu/courses/2-993j-introduction-to-numerical-analysis-for-engineering-13-002j-spring-2005/pages/lecture-notes/	8
5	**Advanced Topics**	Generating Functions: Definitions and basic properties, Applications in counting and solving recurrences. Fuzzy Sets: Definitions and basic properties, Operations on fuzzy sets (union, intersection, complement), Applications and example.	Industry: Financial modelling, operations research. Academia: Discrete mathematics, combinatorial analysis, soft computing, artificial intelligence	6

University of Engineering and Management
Institute of Engineering & Management, New Town Campus

Subject Name: Mental Maths for Professionals **Credit: 1**

Subject Code: MCAGS101 **Lecture Hours: 48**

Module number	Topic	Sub-topics	Mapping with International/National/ State Level Exams	Lecture Hours	Corresponding Assignment
1	**Quantitative Aptitude**	**Textbook: Quantitative Aptitude, Author: R.S Aggarwal, Publisher: S.Chand A. Quant Foundation 1.Number System(Chapter 1)** **2. HCF and LCM (Chapter 2)** **3. Decimal Fractions (Chapter 3)** **4. Simplification (Chapter 4)** **5. Square roots and cube roots (Chapter 5)** **6. Percentage (Chapter 11)-** Basic concept of percentage & its shortcut rules & their applications. **7. Ratio and Proportion (Chapter 13)-** Basic concept of Ratio & Proportion, Shortcut tricks & their	***International Exams*** ***1. GRE (https://www.ets.org/gre/test-takers/general-test/prepare/content/verbal-reasoning.html#accordion-9f58105fc6-item-88093eca37)*** ***National Exams:*** ***1. UPSC Civil Services Exam (https://upsc.gov.in/sites/default/files/Notif-CSP-23-engl-010223.pdf), pg 25-26*** ***2. UPSC Combined Defence Services (https://upsc.gov.in/sites/default/files/Notif-CDS-I-Exam-2023-Engl-211222.pdf), pg 20-21*** ***3. Combined Graduate Level conducted by SSC (https://ssc.nic.in/SSCFileServer/PortalManagement/Upload***	24	1. Assignment on Numerical Problem Solving using Vedic Mathematics principle. 2. Assignment on Numerical Problem-Solving using percentage to fraction relation.

		applications. 8. **Partnership** **(Chapter 14)** concept, rules & Applications, Percentage Advanced problems & shortcuts. **Profit & Loss (Chapter 12)-** Basic concept, formulae, shortcut tricks & their application.	*edFiles/notice_CGLE_03042023.pdf) pg. 20-22* *4. Intelligence Bureau ACIO (https://www.pw.live/exams/wp-content/uploads/2023/11/IB-ACIO-Recruitment-2023-Notification-Emp-News.pdf)* ***State Level Exams:*** *1. Civil Services Executive Exam (WBCS) (https://wbpsc.gov.in/Download?param1=20230225142430_Syllabus.pdf&param2=advertisement, pg 1* *2. Miscellaneous Services Recruitment Examination (file:///C:/Users/UEMK/Downloads/2707970_2019.pdf) pg 1*		
2	**Logical Reasoning**	**Textbook: Modern Approach to Verbal and Non-Verbal Reasoning, Author Dr. R.S Aggarwal, Publisher: S.Chand** **1. Coding and Decoding (Chapter 4)** i. Conditional Coding, ii. Word-Pattern Coding, iii. Chinese Coding, **2. Direction Sense Test(Chapter 8)** i. Direction Sense Test, ii. Direction Distance Test, iii. Shadow based Questions. 3. **Series Completion** (Chapter 1) i. Alphabet Series,	***International Exams*** *1. GRE (https://www.ets.org/gre/test-takers/general-test/prepare/content/verbal-reasoning.html#accordion-9f58105fc6-item-88093eca37)* ***National Exams:*** *1. UPSC Civil Services Exam (https://upsc.gov.in/sites/default/files/Notif-CSP-23-engl-010223.pdf), pg 25-26* *2. UPSC Combined Defence Services (https://upsc.gov.in/sites/default/files/Notif-CDS-I-Exam-2023-Engl-211222.pdf), pg 20-*	24	1. Assignment on Letter Coding, Number Coding, Conditional Coding and Chinese Pattern. 2. Assignment on Directions and Distance 3. Assignment on Indicating based Blood Relation, Coding based Blood Relation and Family Tree based Blood Relation

		ii. Random Series, iii. Number Series, iv. Letter Gap, v. Missing Number Series, vi. Series Completion 4. **Blood Relations** (Chapter 5) – i. Family Tree Questions ii. Indication Type BR, iii. Coding Blood Relations, iv. Miscellaneous Blood Relations.	***21*** ***3. Combined Graduate Level conducted by SSC (https://ssc.nic.in/SSCFileServer/PortalManagement/UploadedFiles/notice_CGLE_03042023.pdf) pg. 20-22*** ***4. Intelligence Bureau ACIO (https://www.pw.live/exams/wp-content/uploads/2023/11/IB-ACIO-Recruitment-2023-Notification-Emp-News.pdf)*** ***State Level Exams:*** 1. ***Civil Services Executive Exam (WBCS) (https://wbpsc.gov.in/Download?param1=20230225142430_Syllabus.pdf&param2=advertisement, pg 1*** ***Miscellaneous Services Recruitment Examination (file:///C:/Users/UEMK/Downloads/2707970_2019.pdf) pg 1***		

Learning Resources: Text Book

1. Quantitative Aptitude- R.S Agarwal
2. Verbal & non-verbal reasoning- R.S Agarwal

University of Engineering and Management
Institute of Engineering & Management, New Town Campus

Syllabus for MCA Admission Batch 2025, 1st Semester

Subject Name: RESEARCH METHODOLOGY AND IPR **Credit: 02**

Subject Code: MCA171 **Lecture Hours: 36 Hrs.**

Name of the Course: **RESEARCH METHODOLOGY AND IPR**	
Course Code: MCA171	**Semester: FIRST**
Duration: 36	**Maximum Marks: 100**
Teaching Scheme	**Examination Scheme**
Theory: 2	End Semester Exam: 100
Tutorial: 1	Continuous Assessment: 100
Credit: 2	

Aim:	
Sl. No.	
1	To develop comprehensive understanding of research methodologies and systematic approaches to academic inquiry
2	To foster awareness of intellectual property rights and their application in research and development
3	To equip students with practical skills for conducting ethical and legally compliant research

Objective:	
Sl. No.	
1	To enable students to formulate well-defined research problems and design appropriate research frameworks
2	To provide knowledge of various data collection methods and their appropriate application in different research contexts
3	To develop competency in data analysis techniques and interpretation of research findings
4	To instil understanding of IPR principles, patent processes, and research ethics in academic and industrial settings
Pre-Requisite: Basic understanding of statistics and data interpretation. Familiarity with computer applications and software tools. Foundational knowledge of research concepts and academic writing	
Course Outcome:	
CO1	Formulate and design a research problem.
CO2	Understand and Comprehend the Data Collection Methods
CO3	Perform Data analysis and acquire Insights.
CO4	Understand IPR and follow research ethics.

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	2	3	2	2	1	1	0	2	1	2	1	3	1	1	3
CO2	2	3	1	3	2	1	0	1	1	1	0	2	1	2	2
CO3	2	2	2	3	2	1	0	1	1	2	1	3	2	3	2
CO4	1	1	1	1	0	2	1	3	1	2	1	2	0	0	2

Module number	Topic	Sub-topics	Lecture Hours
1	**Research Design**	Overview of research process and design, Use of Secondary and exploratory data to answer the research question, Qualitative research, Observation studies, Experiments and Surveys.	6
2	**Data Collection and Sources**	Measurements, Measurement Scales, Questionnaires and Instruments, Sampling and methods. Data - Preparing, Exploring, examining and displaying.	6
3	**Data Analysis and Reporting**	Overview of Multivariate analysis, Hypotheses testing and Measures of Association. Presenting Insights and findings using written reports and oral presentations.	6
4	**Intellectual Property Rights**	Intellectual Property – The concept of IPR, Evolution and development of concept of IPR, IPR development process, Trade secrets, utility Models, IPR & Biodiversity, Role of WIPO and WTO in IPR establishments, Right of Property, Common rules of IPR practices, Types and Features of IPR Agreement, Trademark, Functions of UNESCO in IPR maintenance.	6
5	**Patents**	Patents – objectives and benefits of patent, Concept, features of patent, Inventive step, Specification, Types of patent application, process E-filling, Examination of patent, Grant of patent, Revocation, Equitable Assignments, Licences, Licensing of related patents, patent agents, Registration of patent agents.	6

List of Books Text Books:

Name of Author	Title of the Book	Edition/ISSN/ISBN	Name of the Publisher

1. Cooper Donald R, Schindler Pamela S and Sharma JK, “Business Research Methods”, Tata McGraw Hill Education, 11e (2012).

Reference Books:

1. Catherine J. Holland, “Intellectual property: Patents, Trademarks, Copyrights, Trade Secrets”, Entrepreneur Press, 2007.
2. David Hunt, Long Nguyen, Matthew Rodgers, “Patent searching: tools & techniques”, Wiley, 2007.
3. The Institute of Company Secretaries of India, Statutory body under an Act of parliament, “Professional Programme Intellectual Property Rights, Law and practice”, September 2013.