CHAITANYA SCIENCE AND ARTS COLLEGE

(AUTONOMOUS)

PAMGARH, JANJGIR-CHAMPA (C.G.)

ACCREDIATED "A" GRADE BY NAAC

DEPARTMENT
OF
COMPUTER SCIENCE & APPLICATION

COURSE CURRICULUM & MARKING SCHEME

POSTGRADUATE PROGRAMME

M.SC. COMPUTER SCIENCE

PROGRAM CODE: CCMS06

FIRST & SECOND SEMESTER

Approved By	Board of Studies	Academic Council
Date	30/08/2025	04 SEP 2025

ACADEMIC YEAR 2025-26

SYLLABUS FRAMED ACCORDING TO THE NEP-2020

UNDER THE SCHEME OF CBCS (CHOICE BASED CREDIT SYSTEM)

Phone No:07818299005

Website: https://chaitanyacg.ac.in/ **Email:**principalchaitanya417@gmail.com

Master of Science in Computer Science (M.Sc. Computer Science)

The The **M.Sc. Computer Science** is a two-year postgraduate program that provides advanced knowledge of computer science concepts, programming languages, and modern technologies. It combines theory with practical training, enabling students to design and manage efficient software solutions. The course covers areas such as operating systems, databases, web technologies, and enterprise solutions, with exposure to platforms like **.NET, Java, HTML, C++, and RDBMS.** Graduates are prepared for careers in industry as well as research opportunities.

Programme Outcomes (POs)

After completing the program, students will be able to:

PO-1	Demonstrate an understanding of advanced computing techniques, tools, and their applications.
PO-2	Apply programming knowledge to solve complex problems and contribute to the development of innovative and emerging technologies that address industry and societal needs.
PO-3	Evaluate the impact of technology and provide cost-effective and efficient solutions to end users.
PO-4	Adapt and apply existing design patterns, algorithms, data structures, and techniques to solve real-world problems.
PO-5	Analyze and understand the socioeconomic impact of IT-related solutions.
PO-6	Build a strong foundation for research in future and emerging technological trends.
PO-7	Engage in life-long learning to adapt to the continuously evolving technological environment.
PO-8	Design and develop real-world applications.
PO-9	Understand and apply advanced emerging technologies in real-world applications.
PO-10	Undertake research-based projects and develop commercially viable solutions.

FACULTY OF COMPUTER SCIENCE & APPLICATION
SESSION 2025-26

Proposed Scheme For 2-Year Postgraduate Program

PROGRAMME STRUCTURE: PART	YEAR	SEMESTER	SEMESTER
Part-I	First Year	Semester-I	Semester-II
Part-II	Second Year	Semester-III	Semester-IV

COURSE STRUCTURE:

The syllabus with the paper combination is as under:

SEMESTER-I

SR. NO.	COURSE CODE	TITLE OF COURSE
01.	MCST101	Computer Architecture
02.	MCST102	Advanced Computer Network
03.	MCST103	Java Programming
04.	MCST104	Python Programming
05.	MCSP105 (Practical-1)	Programming Lab in Java
06	MCSP106 (Practical-2)	Programming Lab in Python

SEMESTER -II

SR. NO.	COURSE CODE	TITLE OF COURSE
01.	MCST201	RDBMS using ORACLE
02.	MCST202	Data Structure with C++
03.	MCST203	Software Engineering
04.	MCST204	Cryptography and Network Security
05.	MCST205	Analysis and Design of Algorithm
06	MCSP206 (Practical - 3)	Programming Lab in RDBMS
07.	MCSP207 (Practical - 4)	Programming Lab in Data Structure in C++

SYLLABUS AND MARKING SCHEME FOR THE FIRST & SECOND SEMESTER
SESSION-2025-26
SEMESTER I

Sr. No.	Subject Code	Title	Credits				Marks		Total
			L	T	P	TOTAL	Internal	External	
1	MCST101	Computer Architecture	3	1	-	4	30	70	100
2	MCST102	Advanced Computer Network	3	1	-	4	30	70	100
3	MCST103	JAVA Programming	3	1	-	4	30	70	100
4	MCST104	Python Programming	3	1	-	4	30	70	100
5	MCSP105 (Practical-1)	Programming Lab in JAVA	-	-	2	2			100
6	MCSP106 (Practical-2)	Programming Lab in python	-	-	2	2			100
		Total				**20**			**600**

SEMESTER-II

SR. No.	Subject Code	Title	Credits				Marks		Total
			L	T	P	TOTAL	Internal	External	
1	MCST201	RDBMS using ORACLE	3	1	-	4	30	70	100
2	MCST202	Data Structure with C++	3	1	-	4	30	70	100
3	MCST203	Software Engineering	3	1	-	4	30	70	100
4	MCST204	Cryptography and Network Security (Optional)	3	1	-	4	30	70	100
5	MCST205	Analysis and Design of Algorithm (Optional)	3	1	-	4	30	70	100
6	MCSP206 (Practical - 3)	Programming Lab in RDBMS	-	-	2	2			100
7	MCSP207 (Practical - 4)	Programming Lab in Data Structure in C++	-	-	2	2			100
		Total				**20**			**600**

SEMESTER I

PROGRAMME	**M.SC. COMPUTER SCIENCE – I SEMESTER**	
CORE COURSE	**COURSE CODE: MCST101**	
TITLE	**COMPUTER ARCHITECTURE**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To impart fundamental knowledge of **computer architecture and organization**. To develop the ability to design and construct **cost-effective computer systems**. To provide an understanding of the **basic CPU organization**. To enhance knowledge of **various memory devices** and their applications. To enable students to learn and apply concepts of **I/O communication**	
Course Outcomes	**CO1: Explain** the fundamental concepts of computer architecture and system organization. **CO2: Analyze** the design and functioning of the CPU, including ALU and control unit. **CO3: Compare and select** suitable memory devices as per application requirements. **CO4: Differentiate and apply** various I/O mapping and communication techniques. **CO5: Evaluate** performance issues related to cache memory and virtual memory.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**STRUCTURE OF COMPUTERS:** Computer types, Functional units, Basic operational concepts, Evolution of Computer architecture, Von- Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Data representation, Fixed and Floating point, Error detection and correction codes. COMPUTER ARITHMETIC: Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations.	**15**
UNIT - II	**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. Pipelined processors-Linear pipeline, on linear pipeline- Instruction pipeline design-Arithmetic pipeline design	**15**
UNIT - III	**REGISTER TRANSFER AND MICRO-OPERATIONS:** Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro- Operations, Logic Micro-Operations, Shift Micro- Operations, Arithmetic logic shift unit **MICRO-PROGRAMMED CONTROL:** Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit.	**15**
UNIT - IV	**MEMORY SYSTEM & Buses:** Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID. Cache Memory, Aliasing problem in cache, cache memory mapping techniques-Shared memory organization-Interleaved memory organization, Lower order interleaving, Higher order interleaving. Backplane bus systems-Bus addressing, arbitration and transaction.	**15**

TEXT BOOKS:

M. Moris Mano (2006), Computer System Architecture, 3rd edition, Pearson/PHI, India.

REFERENCE BOOKS:

Carl Hamacher, Zvonks Vranesic, SafeaZaky (2002), Computer Organization, 5th edition,McGraw Hill, New Delhi, India.

William Stallings (2010), Computer Organization and Architecture- designing for performance,8th edition, Prentice Hall, New Jersy.

Anrew S. Tanenbaum (2006), Structured Computer Organization, 5th edition, PearsonEducation Inc,

John P. Hayes (1998), Computer Architecture and Organization, 3rd edition, Tata McGrawHil

SEMESTER I

PROGRAMME	**M.SC. COMPUTER SCIENCE – I SEMESTER**
CORE COURSE	**COURSE CODE: MCST102**
TITLE	**ADVANCED COMPUTER NETWORKING**
Course Credits	**04 Credits**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
No. of Hours Per Lecture	L+T+P (3+1+0)
Course Objective	To develop a clear understanding of the **fundamental concepts of computer networking**. To acquaint students with the **basic taxonomy, models, and terminology** used in computer networking. To introduce **advanced networking concepts** that prepare students for higher-level courses in computer networking.
Course Outcomes	**CO1: Develop** an understanding of the fundamental concepts of computer networking. **CO2: Describe** the basic principles and architecture of computer networking. **CO3: Recognize** and use standard networking terminology and classifications. **CO4: Explain** different types of network models, protocols, and their applications. **CO5: Apply** fundamental and intermediate networking concepts to solve connectivity problems.

Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Computer Networks and the Internet**: History of Computer Networking and the Internet, Internet and Intranet, Network Applications like Web, HTTP, FTP and Electronic Mail in the Internet, Networking Devices, The Network edge, The Network core, Access Networks and Physical media, ISPs and Internet Backbones. Networking Models: 5-layer TCP/IP Model, 7-Layer OSI Model, Internet Protocols and Addressing, Equal- Sized Packets Model: ATM.	**15**
UNIT - II	**Network Routing and its Concepts:** Structure of a Router, Basic Router Configuration, Routing Algorithms: Link-State Routing Algorithm, The distance Vector Routing Algorithm, Hierarchical Routing. Routing in the Internet: Intra-AS Routing in the Internet: RIP, OSPF, BGP. Broadcast and Multicast Routing.	**15**
UNIT - III	**Wireless and Mobile Network:** Introduction, Wireless Links and Network characteristics, CDMA, Wi-Fi 802.11 wireless LANs, Cellular Internet access, Mobility Management: Principles, Mobile IP, Managing mobility in Cellular Networks, Routing calls o mobile users, Handoffs in GSM. Impact of wireless and mobility on higher layer protocols.	**15**
UNIT - IV	**Network Management:** What Is Network Management? The Infrastructure for Network Management, The Internet- Standard Management Framework, Structure of Management Information: SMI, Management Information Base: MIB, SNMP Protocol Operations and Transport Mappings, Security and Administration.	**15**

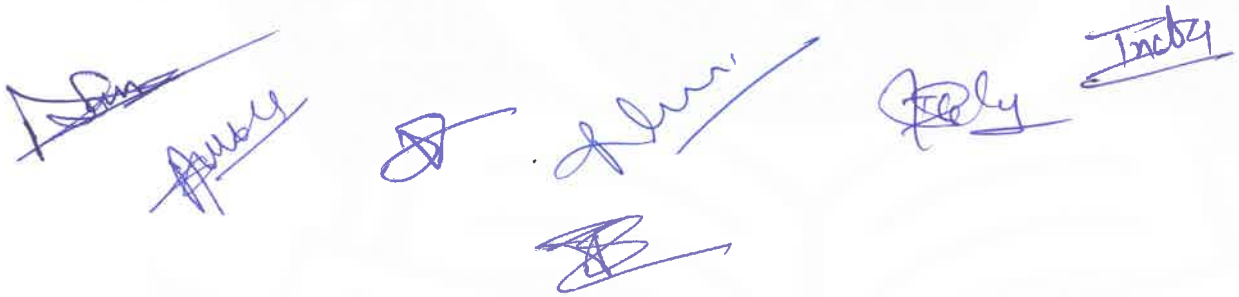
Protocols and Standards.
Describe compare and contrast LAN WAN MAN Intranet Internet AM FM PM and Various Switching Techniques.
Explain the working of Layers and apply the various protocols of OSI & TCP/IP model.
Analyze the Requirements for a Given Organizational Structure and Select the Most Appropriate Networking Architecture and Technologies.
Design the Network Diagram and Solve the Networking Problems of the Organizations with Consideration of Human and Environment.
Install and Configure the Networking Devices.

TEXT BOOKS:

A. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India.
James F. Kurose, Keith W. Ross, Computer Networking a Top-Down Approach, 7th Edition, Pearson, 2001.
Data communications and Networking, Behrouz A Forouzan, Tata Mc Graw-Hill 5th edition, 2013.

REFERENCE BOOKS:

Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.
Kurose, Ross (2010), Computer Networking: A top-down approach, Pearson Education, India.
Larry Peterson and Bruce S Davis "Computer Networks: A System Approach" 5th Edition, Elsevier - 2014
Douglas E Comer, "Internetworking with TCP/IP, Principles, Protocols and Architecture" 6th Edition, PHI - 2014



SEMESTER I

PROGRAMME	**M.SC. COMPUTER SCIENCE – I SEMESTER**	
CORE COURSE	**COURSE CODE: MCST103**	
TITLE	**JAVA PROGRAMMING**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To introduce the **Java SDK environment** for program development and execution. To understand the **fundamentals of Java programming** (variables, data types, control structures, arrays, classes, methods). To learn and implement **OOP concepts** in Java. To explore **strings, vectors, interfaces, packages, and multithreading**. To gain basic knowledge of **file handling, database connectivity, servlets, and web applications**.	
Course Outcomes	**CO1: Explain** and apply object-oriented concepts to solve real-world problems. **CO2: Utilize** the Java SDK environment to create, debug, and execute Java programs. **CO3: Develop** small applications, utilities, and web applications using Java technologies. **CO4: Implement** event handling, layout managers, AWT, Swing, JDBC, and Servlets for software development.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Basics of Java:** History and Basics of Java: Java Environment JDK Tools Java Virtual Machine Java Program Structure Java Language- Tokens Keywords Constants Variables and Data Types. Operators and Expressions Statements - Decision Making Branching and Looping Labeled Loops Statement Jump Statements: Break Continue and Return Command Line Argument	**15**
UNIT - II	**Classes and Objects:** Classes, Objects, defining a Class Adding Variables and Methods Creating Objects Accessing Class Members Constructors Static Members Nesting of Methods Inheritance and Polymorphism: Basics Types Extending a Class Using Super Method Overloading Method Overriding Final Variables and Methods Final Classes Finalize Method Abstract Methods and Classes Visibility Control.	**15**
UNIT - III	**One and Two Dimension Arrays String Array String and String Buffer Classes Vectors Wrapper Classes, Interfaces**: Defining Interfaces Extending Interfaces Implementing Interfaces Accessing Interface Variables Packages: System Packages Naming Conventions Creating Packages Accessing a Package Using Package Adding a Class to a Package Hiding Classes, Exception Handling: Introduction to Exception Handling, Try-Catch Finally Throws Throw, Java Thread Model: Life Cycle of a Thread, Thread Class Runnable Interface.	**15**
UNIT - IV	**Applet Programming:** Creating and Executing Java Applets Inserting Applets in a Web Page Applet Tag Local and Remote Applets Applets Vs. Applications Applets Life Cycle. Database Programming Using JDBC: Introduction to JDBC: JDBC Drivers Types of JDBC Drivers Connecting with Database. J2EE: Introduction of J2EE Web Application Basics Architecture and Challenges of Web Application Servlet Servlet Life Cycle Developing and Deploying Servlets.	**15**

REFERENCE BOOKS:

E. Balagurusamy "Programming with Java a Primer" TMH ISBN- 13: 978-0-07-061713-1 Isbn-10: 0-07-061713-9.

Patrick Naughton and Herbert Schildt "Java: The Complete Reference" TMH Publication ISBN 0-07-463769-X.

Yashavant Kanetkar "Let us Java" BPB Publications.

Ivan Bayross "Web Enabled Commercial Application Development Using HTML DHTML Javascript Perl CGI" BPB Publications

Peter Norton "Java Programming" Techmedia Publications.

Joseph Weber "Using Java 1.2" PHI Isbn-81-203-1558-8.

MCSP105 (PRACTICAL-1) PROGRAMMING LAB IN JAVA

1	JAVA Programming Lab
2	Write a Program in Java to Calculate the Simple Interest.
3	Write a Program in Java to Calculate Sum of Two Numbers Input from Command Line Argument.
4	Write a Program in Java to Calculate Area of Circle Using Scanner Class.
5	Write a Program in Java to Calculate Square Root of a Number.
6	Write a Program in Java to Display Name Age Calendar and Salary of a Person Input from the Keyboard.
7	Write a Program in Java to Display Grading of Student When His Percentage is Input from Keyboard.
8	Write a Program in Java to Display Odd Number from 1 to 100.
9	Write a Program in Java to Display the Following Pattern. 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
10	Write a Program in Java to Calculate the Factorial of a Number.
11	Write a Program in Java to Determine Whether a Number Input from Keyboard is Prime Number or Not.
12	Write a Program in Java to Display the Prime Numbers from 1 to 500 Using Function.
13	Write a Program in Java to Show Accessing Class Members and use a Dot(.).
14	Write a Program in Java to Show Multilevel Inheritance.
15	Write a Program in Java to Show Single Inheritance.
16	Write a Program in Java to Concatenate Two Strings Without Using Library Function.
17	Write a Program in Java to Make First Alphabet Capital of Each Word in a String.
18	Write a Program in Java to Get the Last Index of Any Given Character in a String.
19	Write a Program in Java to Reverse Words of a String.
20	Write a Program in Java to Find Occurrences of Each Character in a String.
21	Java Program to Get String and Count Number of Words in Provided String.
22	Write a Program in Java to Check Given String is Palindrome String or Not in Java.
23	Write a Program in Java to Reverse Each Word of Given String.
24	Write a Program in Java to Get Sub String from a Given String.
25	Java Program to Convert String to Lowercase and Uppercase.
26	Create a Java Applet and Show the use of Drawstring () Function.
27	Create a Java Applet to Show How to use Various Methods of Applet Class and Graphics Class in a Java Applet.
28	Write a Program in Java to Show the use of Interface.
29	Create Two Html Pages with Links to Navigate from One Page to Other Page.
30	Write a Servlet to Display Current Date and Time of Server on Client: Date Servlet
31	Write a Servlet to Display Natural Numbers from 1 to 100: Number servlet
32	JAVA Programming Lab

SEMESTER I

PROGRAMME	**M.SC. COMPUTER SCIENCE – I SEMESTER**	
CORE COURSE	**COURSE CODE: MCST104**	
TITLE	**PYTHON PROGRAMMING**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To provide learners with a solid foundation in Python programming, covering core concepts such as data types, control flow, functions, modules, OOP, file handling, and basic data science tools. By the end, students will be able to write clean, efficient Python code and apply it to solve real-world problems.	
Course Outcomes	**CO1:** Understand Python's history, features, and environment setup to write basic Python programs. **CO2:** Use Python data types, variables, and control flow constructs effectively. **CO3:** Create and manipulate collections like lists, tuples, dictionaries, and sets. **CO4:** Define and invoke functions, use modules, and apply object-oriented programming principles. **CO5:** Handle files, manage exceptions, and perform basic operations with regular expressions. **CO6:** Gain introductory knowledge of multithreading, database connections, and popular data science libraries.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	Introduction to Python including history, features, Python 2 vs 3; installing Python and environment setup; writing first Python program; identifiers, keywords, indentation, comments; command-line arguments and user input; Python data types and variables; core objects, functions, numbers, and mathematical operations; introduction to Python.	**15**
UNIT - II	Control statements including if-else, if-elif-else, while loop, for loop, break, continue, pass, return, assert; lists including creation, operations, iterators, comprehensions, lambda; tuples including ordered sets, operations; ranges and generators; dictionaries including basics, operations; sets including basics, examples.	**15**
UNIT - III	Functions including defining, calling, parameters, return types; anonymous functions (lambda); modules and packages including importing and creating; object-oriented programming (OOP) including classes and objects, constructor, self, variables and namespaces, methods (instance, class, static), inheritance and accessing attributes.	**15**
UNIT - IV	File handling including read, write, append (text and binary); pickle module; exception handling including try-except, try-finally, custom exceptions; introduction to regular expressions; introduction to multithreading and database connection (basics); brief overview of data science libraries including NumPy, Pandas, Matplotlib; IDE and extension setup.	**15**

MCSP106 (PRACTICAL-2) PYTHON PRACTICAL PROGRAM LIST

1. Write a program to compare execution time of list vs tuple operations.
2. Implement a program to check if a number is prime using both iterative and functional (lambda + filter) approaches.
3. Create a program that accepts a string and prints frequency of each character using dictionaries.
4. Simulate a simple command-line calculator that supports addition, subtraction, multiplication, and division using functions.
5. Write a program to generate Pascal's Triangle using nested loops.
6. Implement a program to flatten a nested list without using built-in functions.
7. Demonstrate set operations by writing a program to find students enrolled in: only Python, only Data Science, and both.
8. Write a program to generate all prime numbers within a given range using list comprehensions.
9. Implement a higher-order function that takes another function and a list as arguments and applies it.
10. Write a recursive function to solve the Tower of Hanoi problem.
11. Create a class BankAccount with methods for deposit, withdrawal, and balance inquiry. Add exception handling for insufficient funds.
12. Demonstrate multiple inheritance by creating classes Teacher and Researcher, then a class Professor that inherits from both.
13. Implement a class method and a static method in a class University. Explain the difference with examples.
14. Write a program to read a CSV file and calculate summary statistics (mean, median, std deviation) using Pandas.
15. Implement exception handling for nested try-except blocks and show how finally executes.
16. Write a program that uses regular expressions to extract all valid dates (DD/MM/YYYY) from a text file.
17. Demonstrate multithreading by writing a program that computes square and cube of numbers in parallel.
18. Connect to a database (SQLite/MySQL), create a table students, insert records, and retrieve them.
19. Plot a line chart showing growth of a company over 10 years using Matplotlib.
20. Using NumPy, write a program to generate a random 3×3 matrix and compute its determinant and inverse.

SEMESTER -II

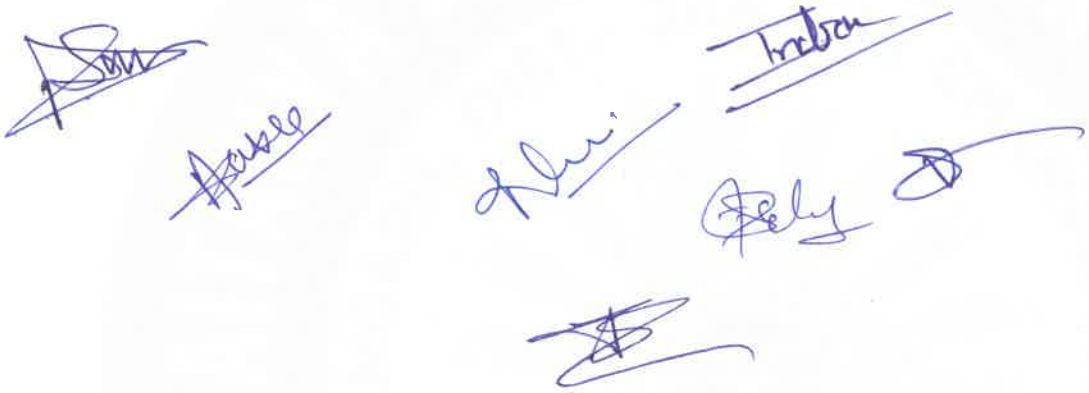
PROGRAMME	**M.SC. COMPUTER SCIENCE – II SEMESTER**	
CORE COURSE	**COURSE CODE: MCST201**	
TITLE	**RDBMS USING ORACLE**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To understand the **fundamental concepts and architecture** of Database Management Systems. To learn **SQL commands** for creating, querying, and updating databases. To study the principles of **transaction management, concurrency control, and recovery**. To develop the ability to **design and implement databases** for real-world applications.	
Course Outcomes	**CO1: Understand** different database models and their applications. **CO2: Write** simple as well as complex SQL queries to retrieve data from databases. **CO3: Perform** data manipulation operations such as insert, update, and delete. **CO4: Apply** functions, procedures, and views for efficient database management. **CO5: Implement** triggers and design relational databases. **CO6: Ensure** data integrity in multi-user and multi-transaction environments.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Overview of Database Management:** Data, Information and knowledge, Increasing use of data as a corporate resource, data processing verses data management, file-oriented approach verses database-oriented approach to data management; data independence, database administration roles, DBMS architecture, different kinds of DBMS users, importance of data dictionary, contents of data dictionary, types of database languages. Data models: network, hierarchical, relational. Introduction to distributed databases.	**15**
UNIT - II	**Relational Model:** Entity - Relationship model as a tool for conceptual design-entities attributes and relationships. ER diagrams; Concept of keys: candidate key, primary key, alternate key, foreign key; Strong and weak entities, Case studies of ER modeling Generalization; specialization and aggregation. Converting an ER model into relational Schema. Extended ER features.	**15**
UNIT - III	**Structured Query Language:** Relational Algebra: select, project, cross product different types of joins (inner join, outer joins, self- join); set operations, Tuple relational calculus, Domain relational calculus, Simple and complex queries using relational algebra, standalone and embedded query languages, Introduction to SQL constructs (SELECT…FROM, WHERE… GROUP BY… HAVING…ORDERBY….), INSERT, DELETE, UPDATE, VIEW definition and use, Temporary tables, Nested queries, and correlated nested queries, Integrity constraints: Not null, unique, check, primary key, foreign key, references, Triggers. Embedded SQL and Application Programming Interfaces.	**15**
UNIT - IV	**Relational Database Design:** Normalization concept in logical model; Pitfalls in database design, update anomalies: Functional dependencies, Join dependencies, Normal forms (1NF, 2NF, 3NF) Boyce Codd Normal form, Decomposition, Multi-Valued Dependencies, 4NF, 5NF. File organization for relational tables, De-normalization. Introduction to Query Processing and Protecting the Database & Data Organizations: Parsing, translation, optimization, evaluation and overview of Query Processing. Protecting the Data Base -Integrity, Security and Recovery. Domain Constraints, Referential Integrity, Assertion, Triggers, Security& Authorization in SQL.	**15**

SEMESTER -II

PROGRAMME	**M.SC. COMPUTER SCIENCE – II SEMESTER**	
CORE COURSE	**COURSE CODE: MCST202**	
TITLE	**DATA STRUCTURE WITH C++**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	This course aims to provide a clear understanding of the fundamental concepts of data structures and algorithms using C programming. It covers the design and utilization of stacks, queues, trees, and graphs, along with an introduction to various searching and sorting techniques.	
Course Outcomes	**CO1: Analyze** algorithms for efficiency and correctness. **CO2: Apply** stack, queue, tree, and graph operations. **CO3: Implement** searching and sorting techniques effectively. **CO4: Develop** problem-solving skills using C programming. **CO5: Apply** data structure concepts in real-world applications.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Introduction to Data Structures:** Definition of Data structure and Abstract data type. Basics of Algorithm. Classification of Data structures: Linear, Non-linear. Arrays: Definition & types of arrays, Memory representation of one- & two-dimensional array, Operations on DS. Linked List: Singly Linked list- Operation on it; doubly linked list- Operation on it; Circular linked list - Operation on it. Overview of C, loops, Functions: call by value and call by reference, Recursive function. Structure: Structure and applications of Structure in various DS. Pointer and applications of Pointer in dynamic memory allocation.	**15**
UNIT - II	**Stacks, Queues:** Stacks; Array representation of stack; Linked representation of stack; Various polish notation's-Prefix, Postfix, infix; Evaluation of a postfix & Prefix expression; Conversion from one another; Application of stack; Queues; Linked representation of queues; Dequeues; Circular queue; Priority queue.	**15**
UNIT - III	**Trees: Binary trees;** Types of binary tree Representation of binary tree in memory; traversing binary tree; Binary search trees; Searching and inserting in binary search trees; Deleting in a binary search, tree; AVL search trees and operation on it. B trees: searching, insertion, deletion; Heap.	**15**
UNIT - IV	Graphs: Terminology & representation; Warshall algorithm; Shortest path; Minimum spanning tree; Kruskal & Dijkstara algorithm; Linked representation of graph; Operation on graph; Traversing a graph. Searching and Sorting: Searching algorithm: linear search, binary search; sorting algorithms: Bubble sort, Insertion sort, Selection sort, Quick Sort, Merge sort and Heap sort.	**15**

References:

- *Let Us C++*, Yashwant Kanetkar, BPB Publications, 10th Edition
- *The C Programming Language*, Kernighan & Ritchie, Prentice Hall
- *Data Structures*, Seymour Lipschutz, McGraw-Hill
- *Data Structures*, Thomas Standish, Addison-Wesley
- *Data Structures Through C*, G. S. Baluja



SEMESTER -II

PROGRAMME	**M.SC. COMPUTER SCIENCE – II SEMESTER**	
CORE COURSE	**COURSE CODE: MCST203**	
TITLE	**SOFTWARE ENGINEERING**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To understand, learn, and apply both theoretical and practical aspects of software development, including paradigms, process models, tools, and techniques. To study the process of software requirements identification, analysis, review, and documentation as per IEEE SRS standards. To gain knowledge of various types and levels of software testing, along with fundamental approaches to test case design. To understand different models of software quality estimation, assurance, and control.	
Course Outcomes	**CO1:** Identify, analyze, and document software requirements using standard tools and methodologies (SRS). **CO2:** Apply project management concepts—planning, scheduling, and risk management for effective software development. **CO3:** Work efficiently in various software development roles such as analyst, architect, programmer, tester, and project manager. **CO4:** Follow coding standards, guidelines, and quality norms to develop reliable software. **CO5:** Design, implement, and optimize test cases for unit, integration, and system-level testing.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Software:** Software Characteristics Components and Applications Software Engineering a Layered Technology Software Development Life Cycle Software Process Models- Linear Sequential Model Prototype & RAD Model Incremental and Evolutionary Process Models. Introduction of Agile Software Development.	**15**
UNIT - II	**Analysis Concept and Principles:** Requirement Analysis, Analysis Principles Requirement Elicitation, Information Gathering Techniques Requirements Specification Requirements, Verification and Validation Requirements Management. Documenting, Software Requirement Specification (SRS) Characteristics of SRS Format of SRS Software Project Planning: Objectives Decomposition Techniques and Empirical Estimation Models. Project Metrics: Software Measurement–Size Oriented Function Oriented Metrics.	**15**
UNIT - III	Design Concepts and Principles: Design Process Design Concepts Design Principles Effective Modular Design Human Computer Interface Design Interface Design Guidelines. System Design: Design Models for Architecture Component Data and User Interfaces; Problem Partitioning Abstraction Cohesiveness Coupling Top Down and Bottom-Up Design Approaches; Functional Versus Object Oriented Approach Design Specification. Coding: Top-Down and Bottom-Up Structure Programming Information Hiding Programming Style and Internal Documentation Verification.	**15**

Faculty

UNIT - IV	**Software Testing:** White and Black Box Testing Levels of Testing UNIT Integration System Testing Functional Testing, Structural Testing, Test Plan Software Testing Strategies, Verification & Validation Incremental & Non-Incremental Testing Top Down and Bottom-Up Integration Testing, Alpha & Beta Testing, White Box and Black Box Testing Techniques, Debugging Techniques. Software Quality: Quality Models, Quality Control and Quality Assurance, ISO, SEI, Capability Maturity Model (CMM). Software Maintenance Need and Categories of Maintenance Software Configuration Management Software Reverse and Reengineering Models.	**15**

REFERENCE BOOKS:

Roger S. Pressman Software Engineering-a Practitioner's Approach McGraw Hill International Edition
K. K. Aggarwal Yogesh Singh Software Engineering
Ian Sommerville- Software Engineering Addison-Wesley Publishing Company
James F. Peter Software Engineering - an Engineering Approach John Wiley
Fairley Richard Software Engineering Concepts Tata McGraw Hill

SEMESTER -II

PROGRAMME	**M.SC. COMPUTER SCIENCE – II SEMESTER**
CORE COURSE	**COURSE CODE: MCSE204**
TITLE	**CRYPTOGRAPHY AND NETWORK SECURITY**
Course Credits	**04 Credits**
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**
No. of Hours Per Lecture	L+T+P (3+1+0)
Course Objective	To develop the skills required for designing and building robust, high-performance, and To learn the concepts of raw sockets and socket programming. To understand the fundamentals of TCP/UDP sockets and simple network management protocols. To study the principles and practices of cryptography and network security. To explore real-world applications and implementations of network security mechanisms.
Course Outcomes	**CO1:** Understand protocols, network interfaces, and design/performance considerations in LANs and WANs. **CO2:** Apply the basics of socket and socket programming for network communication. **CO3:** Analyze contemporary issues in networking technologies and use network tools effectively. **CO4:** Implement client-server programming and apply conventional and public key encryption algorithms. **CO5:** Use message authentication codes, hash functions, digital signatures, and public key certificates for network security.

Syllabus	Description	No. of Lectures / Hours
UNIT - I	**Foundations of Cryptography and security:** Security trends, The OSI Security architecture Security attack, services and mechanism, Ciphers and secret messages, Mathematical tools for cryptography: substitution techniques, modular arithmetic, Euclid's algorithm, finite fields, polynomial arithmetic.	**15**
UNIT - II	**Symmetric Cipher:** Symmetric cipher model, Design Principles of Block Ciphers, Theory of Block, Cipher Design, Feistel cipher network structure, Data Encryption Standard (DES), Strength of DES, Triple DES, Modes of operation. Advance encryption Standard (AES)- Evaluation criteria of AES, AES cipher, key distribution.	**15**
UNIT - III	**Public Key cryptography and Hash function:** Prime numbers and testing for primarily, factoring large, numbers, Principles of public key cryptosystem, RSA algorithm. Key management: Diffie-Helman, Key exchange, Hash and Message authentication Code (MAC), Hash and MAC algorithms, Digital, signature.	**15**
UNIT - IV	**IP and Web security protocols:** Authentication application: Kerberos, Public key infrastructure. E-mail: Pretty Good Privacy (PGP), S/MIME. IP security, Web Security: Secure Socket layer (SSL) and Transport layer security, Secure Electronic Transaction (SET). System Security: Firewall, and Intrusion Detection system (IDS), Malicious Software.	**15**

REFERENCE BOOKS:

Cryptography and Network Security by William Stallings, 4th Edition Pearson Publication
Applied cryptography - protocols and algorithm By Buc Schneier, Springer Verlag 2003
Cryptography and Network Security by Atul Kahate , TMH Publication.
Cryptography and Network Security by Behrouz A. Forouzan, First Edition, TMH Publication.
Network Security: Private Communication in Public World By Charlie Kaufman, Radia Perlman and Mike Speciner, PHI Publication

SEMESTER -II

PROGRAMME	**M.SC. COMPUTER SCIENCE – II SEMESTER**	
CORE COURSE	**COURSE CODE: MCST205**	
TITLE	**ANALYSIS AND DESIGN OF ALGORITHM**	
Course Credits	**04 Credits**	
Total Marks	**Internal Assessment: 30 Marks Maximum Marks: 70 Minimum Passing Marks: 40**	
No. of Hours Per Lecture	L+T+P (3+1+0)	
Course Objective	To introduce students to classic algorithms across various domains and techniques for designing efficient algorithms. To familiarize students with general tools and methods for analyzing computer algorithms. To equip students with the necessary mathematical foundations for algorithm design and analysis.	
Course Outcomes	**CO1:** Check if basic algorithms work correctly and measure how fast they run. **CO2:** Use algorithms and design methods to solve different problems. **CO3:** Compare and understand the difficulty and efficiency of various problems and solutions. **CO4:** Understand and test the correctness and speed of basic algorithms. **CO5:** Use algorithm techniques to solve problems in different areas.	
Syllabus	**Description**	**No. of Lectures / Hours**
UNIT - I	**Introduction:** Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation-Big oh notation, Omega notation, Theta notation and Little oh and omega notation, recurrence elation: Substitution method, Master method Deterministic Algorithms	**15**
UNIT - II	**Divide and Conquer:** General method, applications-Binary search, Quick sort, **Merge sort. Greedy method:** General method, Greedy knapsack problem, **Minimum cost spanning trees:** prims and kruskal's algorithm. Single source **shortest path problem:** Dijkstra's Shortest Path Algorithm, Huffman coding.	**15**
UNIT - III	**Dynamic Programming:** General method, applications- Matrix chain multiplication, optimal binary search trees, Longest Common Sub Sequence Problem. **Back Tracking:** 8- quen problem, Graph Coloring, Hamiltonian Cycle, Branch and Bound: 0/1knapsack problem, travelling sales person problem Non-Deterministic Algorithms.	**15**
UNIT - IV	**Intractable problems:** Basic concepts, non-deterministic algorithms, NP-Hard and NP-Complete problems, Classes P and NP, Reducibility, Satisfiability Problem, Cook"s theorem. Approximation: Graph Coloring, Task scheduling, bin packing, Probabilistic algorithm: Numerical integration, primality testing, Graph Algorithms: BFS and DFS and its applications. Evaluation of Algorithm Lower bound Techniques: Lower bound techniques, Comparison Techniques, reduction.	**15**

REFERENCES:

The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft and J.Ullman, Addison Wesley.
Fundamentals of Computer Algorithms, E. Horowitz and S. Sahani, Galgotia, New Delhi.
Introduction to the Design and Analysis of Algorithms,S.E.Goodman and S.T.Hedetniemi, McGrawHill.
Design Methods and Analysis of Algorithmic, G Brassard and P.Bratley, PHI.
Design Methods and Analysis of Algorithms, S.K.Basu, PHI, 2005.
Introduction toAlgorithms, Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein,MIT Press
Rosen, Kenneth, Discrete Mathematics and Its Applications, McGraw-Hill Science.