School of Computer & Information Sciences

Master of Computer Applications

(w. e. f: 2024-25)

SCHOOL OF COMPUTER & INFORMATION SCIENCES

Vision Statement:

- To invent, create and bring computing technology solutions to the common man, to the privileged and underprivileged sections of India, to bridge the digital divide and eradication of computer ignorance and digital illiteracy and to build a prosperous and technologically advanced nation.

Mission Statements:

MS-1: To pursue academic and research excellence, nationally and internationally

MS-2: To provide training, advisory, and consultancy to all the stakeholders.

MS-3: To lead the efforts in creative and newer modes of instruction delivery & supervision

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications**

Program Educational Objectives (PEOs)

PEO-1: To train the graduates to acquire in depth knowledge of fundamental concepts and programming skills for holistic development

PEO-2: To prepare the graduates for productive careers in software industry, corporate sector and Government Organizations.

PEO-3: To prepare graduates to acquire excellent computing ability so that they can analyze, design and create Solutions for real time problems.

PEO-4: To apply the current tools and techniques to create systems for solving Industry oriented problems.

PEO-5: To prepare graduates to gain multidisciplinary knowledge through real time case studies, projects and industry internship to meet the industry needs.

Mapping Program Educational Objectives (PEOs) with Mission Statements (MS)

	MS-1	MS-2	MS-3
PEO-1	1	3	2
PEO-2	3	2	1
PEO-3	1	2	3
PEO-4	3	2	1
PEO-5	1	2	3

Write ‘3’ in the box for ‘high-level’ mapping, 2 for ‘Medium-level’ mapping, 1 for ‘Low-level’ mapping.

School of Computer & Information Sciences

Name of the Academic Program: <u>**Master of Computer Applications**</u>

Program Outcomes (POs)

PO-1: Computational Knowledge	Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge to conceptualize computing models
PO-2: Problem Analysis	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.
PO-3: Design/Development of Solutions	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO-4: Conduct Investigation of Complex Computing Problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO-5: Model Tool Usage	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.
PO-6: Professional Ethics	Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.
PO-7: Life Long Learning	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.
PO-8: Project Management and Finance	Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO-9: Communication Efficacy	Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
PO-10: Societal and Environmental Concern	Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.
PO-11: Individual Team Work	Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.
PO-12: Innovation and Entrepreneurship	Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Mapping of Program Outcomes (POs) and Program Specific Outcomes (PSOs) with Program Educational Objectives (PEOs)

	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
PO-1	1	2	3	2	1
PO-2	1	3	1	1	2
PO-3	2	2	1	1	3
PO-4	3	3	1	2	1
PO-5	3	2	2	1	1
PO-6	1	2	1	2	3
PO-7	2	3	2	1	1
PO-8	3	1	1	1	2
PO-9	1	1	2	2	3
PO-10	1	2	3	2	1
PO-11	2	2	1	1	3
PO-12	3	2	2	1	1

Mapping of Program Specific Outcomes (PSOs) where applicable.

Write ‘3’ in the box for ‘high-level’ mapping, 2 for ‘Medium-level’ mapping, 1 for ‘Low-level’ mapping.

School of Computer & Information Sciences

Master of Computer Applications

2 Year Program (Revised Curriculum w. e. f: 2024-25)

Course Baskets:

S. No	Name of the Course	L-T-P	Credits	Prerequisites
Basket I: Theory				
1	Theoretical Foundations for Computer Science	4-0-0	4	
2	Computer Organization and Architecture	3-0-0	3	
3	Computer Networks	4-0-0	4	Computer Organization and Architecture
4	Operating Systems	4-0-0	4	Computer Organization and Architecture
5	Algorithms	4-0-0	4	Programming Methodology, Data Structures
6	Software Engineering	3-0-0	3	Object Oriented Programming
7	Optimization Techniques and Statistics for Computer Science	3-0-0	3	
Basket II: Programming				
1	Programming Methodology	3-0-0	3	
2	Data Structures	3-0-0	3	Programming Methodology
3	Object Oriented Programming in Java	3-0-0	3	Programming Methodology
4	Data Base Management Systems	3-0-0	3	
5	Internet Technologies	2-0-2	3	
Basket III: Management				
1	Communication and Technical Writing Skills	3-0-0	3	
2	Software Project Management	3-1-0	4	

Basket IV: Labs				
1	Programming Methodology Lab	0-0-3	1.5	
2	Data Structures Lab	0-0-3	1.5	
3	Object Oriented Programming In Java Lab	0-0-3	1.5	
4	Data Base Management Systems Lab	0-0-3	1.5	
5	Software Engineering Lab	0-0-3	1.5	Object Oriented Programming
6	Python Programming Lab	0-0-3	1.5	
7	Internet Technologies	2-0-2	3	
Basket V: Electives				
1	Mobile Application Development	3-1-0	4	
2	Blockchain Technologies (BCT)	3-1-0	4	
3	Cloud Computing	3-1-0	4	
4	Business Data Analytics	3-1-0	4	
5	Machine Learning	3-1-0	4	
6	Banking Technology and Payment Systems	3-1-0	4	
7	Software Project Management	3-1-0	4	
8	Data Science	3-1-0	4	
Any other elective offered to MCA by the school				

NEP Baskets:

DSC: Department Specific Core

SSC : School Specific Core

FSC: Faculty Specific Core

ScSE: School Specific Elective

Int/ RI/ CE: Internship/ Research Internship/ Community Engagement

Revised Curriculum: **(w. e. f: 2024-25)**

Course Code	Name of the Course	Course Basket	NEP Basket	L-T-P	Contact Hours	Credits
FIRST SEMESTER						
CA411	Theoretical Foundations for Computer Science	Theory	DSC	4-0-0	4	4
CA412	Programming Methodology	Programming	DSC	3-0-0	3	3
CA413	Computer Organization and Architecture	Theory	DSC	3-0-0	3	3
CA514	Data Base Management Systems	Programming	DSC	3-0-0	3	3
CA517	Communication Skills and Technical Writing	Management	FSE	3-0-0	3	3
CA415	Programming Methodology Lab	Labs	DSC	0-0-3	3	1.5
CA516	Data Base Management Systems Lab	Labs	DSC	0-0-3	3	1.5
CA417*	Optimization Techniques and Statistics for Computer Science	Theory	SSC	3-0-0	3	3
					Total Credits	**22**
SECOND SEMESTER						
CA462	Operating Systems	Theory	DSC	4-0-0	4	4
CA463	Data Structures	Programming	DSC	3-0-0	3	3
CA464	Object Oriented Programming in Java	Programming	DSC	3-0-0	3	3
CA467*	Internet Technologies	Programming + Labs	DSC	2-0-2	4	3
	Elective - 1	Electives	ScSE	3-1-0	4	4
CA465	Data Structures Lab	Labs	DSC	0-0-3	3	1.5
CA466	Object Oriented Programming in Java Lab	Labs	DSC	0-0-3	3	1.5

Total Credits						**20**
After end of the Second Semester, before going for the summer vacation, students need to complete the registration for the project						
THIRD SEMESTER						
CA512	Software Engineering	Theory	DSC	3-0-0	3	3
CA511	Algorithms	Theory	DSC	4-0-0	4	4
CA461	Computer Networks	Theory	DSC	4-0-0	4	4
	Elective – 2	Electives	ScSE	3-1-0	4	4
	Elective – 3	Electives	ScSE	3-1-0	4	4
CA515	Software Engineering Lab	Labs	DSC	0-0-3	3	1.5
CA518*	Python Programming Lab	Labs	SSC	0-0-3	3	1.5
CA558*	Project Evaluation – 1	Project	Int/RI/CE			6
Total Credits						**28**
FOURTH SEMESTER						
CA559*	Project Evaluation - 2	Project	Int/RI/CE			6
CA561	Internship	Internship	Int/RI/CE			6
Total Credits						**12**
1. There is no requirement of submitting a report for Internship **2. No marks for Internship, only Satisfactory or Unsatisfactory** **3. For project final evaluation, students need to submit a report** **Project Evaluation – 1: Requirement Specification, Research and Design** **Project Evaluation – 2: Implementation, Report and Demo** ***: New courses, and new course codes assigned.**						
Total Credits: 82						

School of Computer & Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-I)**

Course Code : CA411 Title of the Course : Theoretical Foundations of Computer Science

L-T-P :. 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a mathematics course at 10+2 level

Course Outcomes (COs) (5 to 8)

After completion of this course successfully, the students will be able to……

- CO1: Apply predicate and propositional logic to represent and solve problems. (Apply)
- CO2: Discuss various ways of simplification and apply the same on minimizing logical circuits, (Understand)
- CO3: Using principle of recursion, be able to frame a real-world situation as a recurrence relation and solve. (Apply).
- CO4: Describe counting principles (Understand)
- CO5: Apply counting principles in real world scenarios. (Apply)
- CO6: Describe graphs and trees techniques (Understand)
- CO7: Apply the graphs and trees techniques to solve the real time problems (Apply)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	3	2	1	1	1	1	1	1	1
CO2	1	1	1	1	2	1	2	1	1	1	1	3
CO3	1	1	1	1	2	1	1	3	1	1	1	1
CO4	1	1	2	1	1	1	1	1	3	1	1	1
CO5	1	1	3	1	1	1	1	1	1	2	1	1
CO6	1	3	1	2	1	1	1	1	1	1	1	1
CO7	1	1	1	1	2	1	1	1	1	3	3	1

Detailed Syllabus:

UNIT-I: Sets, Relations and Functions

Sets, relations and functions; Methods of proof; Equivalence relations; Cardinality; Countable and uncountable sets

UNIT-II: Introductory Logic

Fundamentals of Logic; Logic operators such as AND, OR etc., Truth tables; Logical inferences; Methods of proofs of an implication; First order logic; Predicate calculus Predicates and Quantifiers; Rules of inference for quantified propositions

UNIT-III: Recurrence Relations:

Recursion, Forming and solving recurrence relations by substitution method and generating functions; Method of characteristic roots; solving inhomogeneous recurrence relations

UNIT-IV: Boolean Algebra:

Partial order relations; Lattices; Boolean algebra; Combinatorial circuits; Minimization of Boolean functions using Karnaugh maps

UNIT-V: Theory of Graphs

Graphs, subgraphs, isomorphism, proofs; Types of graphs; paths and cycles; Adjacency matrices; Transitive closure; Connectivity; Directed acyclic graphs; Planar graphs and Euler's formula; Dual of a graph; Hamiltonian and Eulerian graphs; Applications like matching and colouring graphs; Graph traversals (BFS and DFS); Trees; Spanning trees.

Reference Books:

1. Kenneth H Rosen (2012), "*Discrete Mathematics and Its Applications*",7th Edition, McGraw Hill, NY
2. Ralph P Girimaldi(2003), "*Discrete and Combinatorial Mathematics –An Applied Introduction*", 5th Edition, Pearson Addison Wesley, Indian Edition
3. J.R Mott, A Kandel, T.P Baker (2015), "*Discrete Mathematics for Computer Scientists and Mathematicians*", Pearson
4. Ronald L Graham, Donald E Knuth, Oren Patashnik(1994), "*Concrete Mathematics- A Foundation of Computer Science*", 2nd Edition, Addison Wesley .
5. Susanna S. Epp(2010), "*Discrete Mathematics with Applications*", 4th Edition, Brooks/Cole Cengage Learning.

School of Computer & Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-I)**

Course Code : CA412/CA415 Title of the Course : Programming Methodology & PM Lab

L-T-P : 3-0-3 Credits : 3+1.5 (4.5)

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Create specification from problem requirements by asking questions to disambiguate the requirement Statement. (Create)

CO-2: Design the solution from specification of a problem and write pseudo code of the algorithm. (Create)

CO-3: Analyze algorithms by tracing algorithms with test cases. (Analyze)

CO-4: Develop C programs using all supported features and compile them using Makefile. (Create)

CO-5: Analyze programs using debugging tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	3	1	1	1	1	1	1	1
CO2	1	2	1	1	3	1	1	2	1	1	1	1
CO3	1	2	3	1	1	1	1	1	1	1	1	1
CO4	1	1	1	2	1	1	1	1	1	1	1	3
CO5	1	2	1	1	3	1	1	1	1	1	1	1

.

Detailed Syllabus:

UNIT-I**:** Introduction to problem solving:
Problems and problem instances; Informal approach to program design: generalisation, special cases, and algorithms, breaking down a problem into functions, input and output.

UNIT-II**:** Introduction to the 'C' programming language:
Program structure; main () function; unnamed and named blocks; basic data types, variables, declaration and definition; initialisation and assignment; arithmetic operators and precedence; implicit and explicit type conversions; arrays; boolean variables and logical operators.

UNIT-III**:** Control structures:
Branching and iteration; functions and parameters; break (), return () and exit () functions; local and global variables; function prototypes.

UNIT-IV: Pointer variables and dynamic structures:
Static and dynamic (run-time) memory structures; static variables; breaking a program across multiple files; creating and linking libraries.

UNIT-V: Detecting and correcting common errors:
Debugging and debuggers; documenting programs; good programming practices; programming exercise (writing a program of at least 200 lines split across multiple files).

Reference Books:

1. Brian W. Kernighan, Dennis M. Ritchie. "The C Programming Language, 2nd Edition", Prentice-Hall India.
2. G. Michael Schneider. "Introduction to Programming and Problem Solving with PASCAL", John Wiley and Sons.
3. Paul Deitel and Harvey Deitel . "C How to Program", Pearson Education India.
4. Stephen Kochan. "Programming in C", Pearson Education India.
5. Brian W. Kernighan and R. Pike. "The Unix Programming Environment", Prentice-Hall India.
6. Chakravarthy Bhagvati. "How to Program (An Informal Guide)", https://scis.uohyd.ac.in/~chakcs/howtoprogram.pdf

School of Computer & Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-I)**

Course Code : CA413 Title of the Course : Computer Organization & Architecture

L-T-P :. 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done a programming course at any level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO1: Demonstrate arithmetic operations and assess their performance (Apply).
- CO2: Describe basic Instruction Set Architecture (ISA) (Understand)
- CO3: Explain the basic pipelining of instructions (Understand)
- CO4: Examine how the memory hierarchy has impact on performance of software. (Analyze)
- CO5: Describe Interrupt handling and DMA access for performing I/O. (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1	1	1	1	1	1	1	1	1	1
CO2	1	1	2	1	3	1	1	1	1	1	1	1
CO3	1	2	3	1	1	1	1	1	1	1	1	1
CO4	3	1	1	1	2	1	1	1	1	1	1	1
CO5	1	1	3	1	2	1	1	1	1	1	1	1

Detailed Syllabus:

UNIT - I: Computer Evolution & Arithmetic:

A Brief History of computers, Designing for Performance, Von Neumann Architecture, Hardware architecture, Computer Components, Interconnection Structures, Bus Interconnection, Scalar Data Types, Fixed and Floating-point numbers, signed numbers, Integer Arithmetic, 2's Complement method for multiplication, Booths Algorithm, Hardware Implementation, Division, Restoring and Non-Restoring algorithms, Floating point representations, IEEE standards, Floating point arithmetic

UNIT - II: The Central Processing Unit:

Machine Instruction characteristics, types of operands, types of operations, addressing modes, Instruction formats, Instruction types, Processor organization, Processor as running example, Programmers model of, max/min mode, Register Organization, Instruction cycles, Read Write cycles, assembly instruction examples to explain addressing modes

UNIT – III: The Control Unit:

Single Bus Organization, Control Unit Operations: Instruction sequencing, Micro operations and Register Transfer. Hardwired Control: Design methods – State table and classical method, Design Examples - Multiplier CU. Micro-programmed Control: Basic concepts, Microinstructions and micro- program sequencing

UNIT - IV:

Memory Organization: Characteristics of memory systems, Internal and External Memory, Types of memories: ROM: PROM, EPROM, EEPROM, RAM: SRAM, DRAM, SDRAM, RDRAM, High-Speed Memories: Cache Memory, Organization and Mapping Techniques, Replacement Algorithms, Cache Coherence, Virtual Memory: Main Memory allocation, Segmentation, Paging, Address Translation Virtual to Physical. Secondary Storage: Magnetic Disk, Tape, DAT, RAID, Optical memory, CDROM, DVD

UNIT - V: I/O Organization:

Input/ Output Systems (features and principles), Programmed I/O, Interrupt Driven I/O, Interrupt structure, Direct Memory Access (DMA), features Buses and standard Interfaces: Synchronous, Asynchronous, Parallel I/O features, Serial I/O features, PCI, SCSI, USB Ports Working mechanisms of Peripherals: Keyboard, Mouse, Scanners, Video Displays, Touch Screen panel, Dot Matrix, Desk-jet and Laser Printers.

UNIT - VI:

Case Studies: Concepts RISC: Instruction execution characteristics, RISC architecture and pipelining. RISC Vs CISC. ARM and Embedded Systems PowerPC, Intel X86 Evolution from 32bit to 64bit architectures. AMD Opteron

Reference Books

1. Patterson D.A. & Hennesy J.L., *Computer Organisation & Design: The Hardware/Software Interface*.
2. Computer Organization and Architecture, 10/E William Stallings ISBN-10: 0134101618 • ISBN-13: 9780134101613- See more at: http://www.pearsonhighered.com/pearsonhigheredus/educator/product/products_detail.page?isbn=9780134101613&forced_logout=forced_logged_out#sthash.WVVJbZUb.dpuf

School of Computer and Information Sciences

Name of the Academic Program: MCA (MCA-I)

Course Code: CA514 Title of the Course: **Database Management Systems**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Understand the basic concepts of database management systems (Understand)

CO-2: Design the database systems using ER and EER Models (Create)

CO-3: Use SQL to query the database systems (Apply)

CO-4: Evaluate the database design aspects by considering normalization principles (Evaluate)

CO-5: Explain indexing and hashing mechanisms used in database systems (Understand)

CO-6: Understand the transaction management in database systems (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	2	1	2	2	2	1	1	1	2
CO2	3	2	1	2	3	1	2	2	2	1	1	3
CO3	2	1	2	1	3	1	2	1	1	2	2	2
CO4	1	2	3	2	3	1	1	3	1	2	3	2
CO5	2	2	3	1	1	2	3	1	1	2	1	1
CO6	3	1	2	1	1	1	2	1	2	3	2	1

Detailed Syllabus:

UNIT 1: Introduction: Data and Information, Overview of File Management System, Database Systems Versus File Systems, Levels of Data Abstraction, Data Models, Database Languages- Data Definition Language, Data Manipulation Language, Data Control Language, Transaction Control Language, Functions of Database Administrator, Taxonomy of Database Users. Database System structure

UNIT 2: Data Models: EntityRelationship model- Basic Concepts, Constraints, Keys, Design Issues, E-R Diagram, Weak Entity Sets, Extended E-R Features, Design of E-R Database Schema, Reduction of E-R Schema to tables. Relational Model- Structure of Relational Database, Normalization, Functional Dependencies, Decomposition, Desirable properties of Decomposition, Normal Forms, Denormalization for Performance

UNIT 3: Structured Query Language (SQL). Basic structure, Set operations, Aggregate functions, Null Values, Nested Subqueries, Views, Complex queries, joins Modification of database schema, Integrity Constraints. Relational Algebra & Relational Calculus.

UNIT 4: Application Design & Development: Discussion on complex data types like semi structured data, object based data, temporal data and spatial data. Big data analytics and how those applications place distinct demand on data management compared with traditional database applications (NoSQL, MapReduce, Apache Spark, streaming data and graph databases.

UNIT 5: Storage and File Structures: File organization, Organization of Records in Files, Data Dictionary Storage. Indexing and Hashing: Basic Concepts, Ordered Indices, Static Hashing, Dynamic Hashing, Comparison of Ordering Indexing and Hashing, Index Definition in SQL, Multiple-Key Access

Transaction Management: ACID properties, Transaction State, Concurrent Executions, Serializability, Recoverability, Transaction Definition in SQL, Concurrency Control – Lock-Based Protocols, Timestamp-Based Protocols.

Reference Books:

1. A. Silberschatz, H. F. Korth and S. Sudarshan, *Database System Concepts*, 7th Edition McGrawHill Publications
2. R. Elmasri, S. B. Navathe: *Fundamentals of Database Systems*, 7th Edition, Pearson Publication.
3. Hector Garcia-Molina, Jeffrey D Ullman, Jennifer Widom: Database Systems The Complete Book, Second Edition, Pearson, Prentice Hall.

School of Computer and Information Sciences

Name of the Academic Program: MCA (MCA-I)

Course Code: CA516 Title of the Course: **Database Management Systems Lab**

L-T-P : 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

CO-1: Design and implement database schema using RDBMS concepts (Create)
CO-2: Demonstrate physical and logical data independence (Apply)
CO-3: Use SQL to query the database systems (Apply)
CO-4: Demonstrate procedural data manipulation language constructs (Apply)
CO-5: Develop an application using SQLite and Duckdb features (Create)
CO-6: Develop a database application (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	3	2	2	1	3	2	2	1	1	2
CO2	2	2	3	1	2	1	1	2	1	1	1	1
CO3	2	3	3	2	3	1	2	2	1	2	2	2
CO4	3	2	3	2	3	1	2	2	1	2	3	2
CO5	3	3	3	3	2	1	2	3	1	2	2	2
CO6	3	3	3	3	2	1	2	3	1	3	2	2

Detailed Syllabus:

- Introduction to SQL. Features of SQL, DDL Statements and DML commands. Writing simple SQL queries using DDL statements and DML commands

- Introduction to inner, outer and natural joins. Writing nested queries and correlated nested queries to retrieve and update the data. Writing SQL queries using EXISTS, NOT EXISTS, explicit join operation, aggregate functions, group by and having clauses.

- Creating virtual tables (views). Using views in SQL queries.

- PL/SQL programming.

- Getting started with sqlite3 : https://sqlite.org/cli.html#getting_started /

- Includes changing output formats, querying the database schema, opening database files, redirecting I/O, Accessing zip archives as database files, converting an entire database to a text file, Index recommendations (SQLite Expert), working with multiple database connections etc.

- Working with DuckDb : https://duckdb.org/

- Loading dataset into DuckDb using the SQLite Scanner.

- Mini project.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications

Course Code: CS517 **Title of the Course**: Communication Skills and Technical Writing

L-T-P: 3-0-0 **Credits**: 3

Prerequisite Course / Knowledge (If any): Nill

Course Objective:

Understanding the vital role of the English language as a medium for global communication in the field of technology, this course is tailored to enhance the technical and communication skills of students. The course focuses on refining essential skills such as reading, writing, listening, and speaking, which are critical for both academic and professional success. By incorporating relevant examples from the IT and computer science fields, students will be exposed to various forms of technical and professional communication. Additionally, self-guided tasks are designed to help students sharpen their communication abilities, preparing them to effectively engage in a modern, globalized tech environment.

COURSE OUTCOMES

CO1: Understand and Organize Ideas Effectively (Apply)

(Individual & Group Presentations; Structured Writing Exercises)

CO2: Analyze and Engage in Debates (Apply)

(Debates; Speech Practice)

CO3: Exhibit effective group communication skills in discussions (Apply)

(Group Discussions; Collaborative Case Studies)

CO4: Employ interview techniques with confidence (Apply)

(Mock Interviews; Question & Answer Sessions)

CO5: Develop and Apply Technical and Review Writing Skills (Apply)

(Technical Report Writing; Literature Reviews)

CO6: Create and Deliver Effective Oral Presentations (Create)

(Individual and Group Presentations; Public Speaking Drills)

CO7: Cultivate active listening and interpretation skills (Analyze)

(Listening Drills; Audio-Based Group Discussions; Speech and Lecture Interpretation)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	2	1	1	2	3	3	3	3
CO2	1	1	1	1	1	1	2	3	1	3	3
CO3	1	1	1	1	1	1	2	3	2	3	3
CO4	1	1	1	1	1	1	2	3	1	3	3
CO5	1	1	1	1	1	2	3	3	1	3	3
CO6	1	1	1	1	1	2	3	3	3	3	3
CO7	1	1	1	1	1	2	3	3	3	3	3

DETAILED SYLLABUS

UNIT-I: Non-Verbal Communication

Body language in group discussions and interviews; Roleplay exercises to practice non-verbal cues and responses.

UNIT-II: Vocabulary Building

Synonyms, antonyms, root words, one-word substitutes, prefixes, suffixes, etymology, analogy, idioms, and phrases.

UNIT-III: Reading Comprehension and Writing Skills

Techniques for reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, and critical reading. Resume and CV writing; e-writing and technical writing; effective presentation of research; data collection, organization, and analysis.

UNIT-IV: Speaking Skills and Group Dynamics

Group discussions focusing on listener roles, body language, summarizing, and fluency; Presentation skills to overcome fear and build confidence; Oral presentations (individual and group) and written presentations (posters, reports, assignments).

UNIT-V: Interview Skills

The interview process (before, during, and after); Pre-interview planning; Opening strategies; Answering techniques; Interviewing via telephone and videoconferencing.

Reference Books:

1. "Technical Communication" by Meenakshi Raman & Sangeeta Sharma, Oxford University Press, 2009

This book provides a comprehensive foundation in technical communication, covering essential writing and presentation skills. It will be useful for teaching CO1 (Organizing Ideas Effectively), CO5 (Technical and Review Writing), and CO6 (Creating and Delivering Presentations).

2. "Communications Skills" by Leena Sen, PHI Learning Pvt. Ltd., New Delhi, 2009

Focuses on developing effective communication strategies, both verbal and non-verbal. It will help with CO2 (Engaging in Debates), CO3 (Group Communication Skills), and CO7 (Active Listening and Interpretation Skills).

3. "Handbook for Technical Writing" by David A McMurray & Joanne Buckley, CENGAGE Learning, 2008

This is an excellent resource for CO5 (Technical Report Writing) and CO4 (Interview Techniques), offering practical guidelines on writing clear, concise, and well-organized technical documents.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA)

Course Code: CA417 Title of the Course: Optimization Technique and Statistics for Computer Science

L-T-P: 3-0-0 Credits: 3

Prerequisite Course / Knowledge (If any): Nill

Course Objective: These subject deals various optimization techniques for linear programming, Transportation, Assignment Problems, PERT and CPM. It also deals with the fundamental statistical and probability methods like data types, sampling techniques, data visualization and presentation, descriptive statistics and different types of distributions

Goal: To learn about managerial concepts like statistical decision making, optimization etc.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Understanding the data collection and sampling techniques (Understand)

CO-2: Understanding different data distribution techniques and descriptive statistical methods (Understand)

CO-3: Understanding various mathematical applications in industries. (Understand)

CO-4: Decision making for real-time environment. (Apply)

CO-5: Formulating Real-world problems as optimization problems (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	2	1	1	2	1	1	1	1	1	1	1
CO2	3	1	1	2	1	1	1	1	1	1	1
CO3	1	1		3	2	1	1	1	1	1	1
CO4	1	1	2	1	1	1	2	1	1	1	1
CO5	3	2	3	3	3	1	1	1	1	1	1

Unit-I:

Introduction: Data Collection and Descriptive Statistics, Inferential Statistics and probability Models, Population and Samples. Data: quantitative and qualitative, attributes, variables, scales of measurement nominal, ordinal, interval and ratio. Presentation: tabular and graphical, including histogram and ogives, consistency and independence of data with special reference to attributes.

Descriptive Statistics: Describing Datasets, Single Point Summarization, Paired Datasets.

Measures of Central Tendency: mathematical and positional. Measures of Dispersion: range, quartile deviation, mean deviation, standard deviation, coefficient of variation, Moments, absolute moments, factorial moments, skewness and kurtosis.

Unit-II

Probability: Sample Space and Events, Axioms of Probability, Conditional Probability. Random Variables and Expectations: Random variables, Jointly Distributed Random variables, Expectation, Variance, Co-variance, Probability Distributions. Parameter Estimation-Maximum Likelihood Estimates; Regression Analysis; Applications, Markov Process, Poisson Process.

Unit-III:

Linear Programming - Mathematical Model assumption of linear Programming – Graphical method - Principles of Simplex method, Big-M Method, Duality, Dual simplex method.

Unit IV:

Transportation and Assignment problem - Integer Programming, Branch-and-Bound Techniques.

PERT & CPM - Network representation - backward pass - Forward pass -computation – PERT Network

Books:

1. S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, Elsevier.
2. W. Feller, An Introduction to probability Theory and its Applications- Vol. 1, Wiley.
3. Operations Research, Manmohan, P.K. Gupta, Kanthi Swarup, S. Chand & Sons - 1997.

1. Operations Research, Hamdy A Taha, Pearson Education, 7th edition, 2002
2. Problems in Operations Research, P.K. Gupta, D.S. Hira, S. Chand Publishers

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA462 Title of the Course: Operating Systems

L-T-P : 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): -- None

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Discuss the ways system calls work. (Understand)
- CO2: Develop basic process management tasks such as scheduling, deadlock avoidance algorithms. (Create)
- CO3: Develop paging algorithm. (Create)
- CO4: Construct simple device drivers. (Create)
- CO5: Describe different file systems in existence and learn the pros and cons of the various systems. (Understand)
- CO6: Examine real world OS scheduling algorithms such as those used in Linux and Windows. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	1	2	3	1	1	1	1	1	1	1	1
CO2	1	2	3	1	1	1	1	1	1	1	1	1
CO3	1	1	1	3	2	1	1	1	1	1	1	1
CO4	1	1	3	1	2	1	1	1	1	1	1	1
CO5	1	3	2	1	1	1	1	1	1	1	1	1
CO6	1	1	1	1	1	1	1	1	1	2	1	3

Detailed Syllabus:

UNIT - I: Introduction and Operating System Structures

Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT - II: Process and Thread Management

Process Concept, Process operations, Process Scheduling, Extended Process State Diagram, Process Context Switch in detail; Inter process Communication: Pipes, Named Pipes, Shared Memory; Process Synchronization: Signals, Mutexes, Semaphores, Monitors; Thread Management: thread creation, thread scheduling, thread synchronization; Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock.

UNIT - III: Memory Management

Memory allocation techniques: paging and segmentation, Swapping, structure of the page table; Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB).

UNIT - IV: File System Management

Disk management: formatting, boot block, swap-space management, RAID structure; Disk scheduling algorithms: elevator, C-SCAN; File concept, Access methods, Directory structure, file sharing, protection, file system structure; file system implementation: file system metadata storage structures such as inode, allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT - V: I/O Management

I/O devices: polling, interrupt-driven, DMA; Application I/O interface: character and block devices, network devices; clocks and timers, nonblocking and asynchronous I/O, vectored I/O; Kernel I/O interface: I/O scheduling, Buffering, Caching.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts, 9th edition, Wiley.
2. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
3. W. Richard Stevens,. Advanced Programming in Unix Environment, Pearson Education.
4. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
5. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
6. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
7. Robert Love. Linux Kernel Development, Pearson Education.
8. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA463 Title of the Course: Data Structures

L-T-P: 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1 Show and demonstrate which data structures are used for static and dynamic allocations. (L2)
- CO-2 Solve the problem where in elements can be traversed by either direction and select the suitable data structure for this idea using C/Java Programming Language (L3)
- CO-3 Analyze the time taken to solve the problem by using C/Java programming language (L4)
- CO-4 Assess the solution in terms of efficiency, modularity and well-documented programs in C/Java under Linux environment (L5)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	1	1	1	1	1	2	2
CO2	3	3	3	3	3	1	1	1	1	1	2	2
CO3	3	3	3	3	3	1	1	1	1	1	2	2
CO4	3	3	3	3	3	1	1	1	1	1	2	2

Detailed Syllabus

Unit-I

Introduction to data structures and data types: Primitive and Non-Primitive types, Arrays- Sparse matrix, Stacks, Queues, Circular queues, Priority queues, Dequeues, Conversions and Evaluations of expressions, Polynomial representation using arrays, Time complexity analysis of algorithms with respect to data structure operations

Unit-II

Linked Lists: Linked stacks and queues, Circular and Doubly linked lists, Polynomial representation using linked lists.

Unit-III

Trees and Graphs: Binary Trees, Tree Traversal, Binary Search trees and basic operations, Heaps, AVL Trees, height balanced trees. , Graphs – Representation of the graphs, Graph Traversals.

Unit-IV

Sorting and Searching Mechanism: Selection sort, Bubble sort, Insertion sort, Linear Search, Binary Search, Hash Tables.

Reference Books:

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.

2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.

3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.

4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.

8. Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.

9. The *C Programming* Language by Brian W.Kernighan , Dennis M. *Ritchie*

10. *Michael J. Folk and Bill Zoellick, “File Structures” (Second Edition).*

11. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition Addison- Wesley, 1997.

12. Schaum‟s Outline Series, “Data Structure”, TMH, Special Indian Ed., Seventeenth Reprint, 2009.

13. Mary E. S. Loomes, “Data Management and File Structure”, PHI, 2nd Ed., 1989.

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-II)**

Course Code: CA465 Title of the Course: Data Structures Lab

L-T-P: 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1 Solve a problem by choosing appropriate data structures in C programming language (L2)
- CO-2 Select suitable data structure for an idea and propose solution using C Programming Language (L2)
- CO-3 Analyze the time taken to solve the problem by using C programming language (L4)
- CO-4 Assess the solution in terms of efficiency, modularity and well-documented programs in C under Linux environment (L5)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	1	1	1	1	1	2	2
CO2	3	3	3	3	3	1	1	1	1	1	2	2
CO3	3	3	3	3	3	1	1	1	1	1	2	2
CO4	3	3	3	3	3	1	1	1	1	1	2	2

Detailed Syllabus

Unit-I

Implementation of Stacks and different types of Queues data structures using arrays, Conversions and Evaluations of an expressions.

Suggested Assignment

Large integer arithmetic. Arithmetic operations are to be performed on very large integers of N digits (where 0 < N < 20). Write C program that performs the operations of multiplication and division on such large integers.

Two stacks using single array. Implement two stacks using a single array such that neither overflow unless total number of elements in both the stacks is equal to the size of the array.

Deque. Deque is a queue which allows insertions and deletions at both ends. Write a C program that implements deque using both *arrays* and *linked list*. Each implementation should provide support for basic operations such as (i) Insertion of an element at both ends (ii) Deletion of an element from both ends (iii) Displaying all elements of deque.

Evaluation of arithmetic expression. Write a C program that reads an input arithmetic expression in (INFIX, PREFIX, POSTFIX) notations and outputs its result.

Arithmetic expression notation conversion. Write a C program that reads an input arithmetic expression in infix notation (fully parenthesized) and converts it into an output arithmetic expression in postfix notation.

Unit-II

Implementation of basic Linked List operations such as addition, updation, deletion, searching and traversal of all elements of the list

Suggested Assignments

Implement Stack data structure (using linked list) operations. Write C program to perform the basic operations on the stack.

Implement Queue data structure (using linked list) operations. Write C program to perform the basic operations on the queue.

Doubly Linked List operations. Linked list (doubly) is an important data structure for dynamic allocation wherein elements can be traversed by either direction. Write C program to perform the basic operations on the linked list.

Union and Intersection. Given two linked lists of numbers, write a program that finds a resultant linked list which is union of the two input linked lists and another resultant linked list which is intersection of the two input linked lists.

Detecting cyclic linked list. Write a C program that detects whether a given linked list is cyclic or not, if yes, then return the node where the cycle begins.

Unit-III

Implementation of Trees and Graphs of basic operations. Implementation of Adjacency Matrix and List Representation. Breadth and Depth First Search

Suggested Assignments

Tree traversals. Write C program to display a tree using all the methods of traversals: (i) Inorder traversal, (ii) Preorder traversal, (iii) Postorder traversal

Binary Search Tree. Binary Search Tree is an important data structure for dynamic allocation and optimized searching. Write C program to perform the basic operations on binary search tree (BST): (i) Adding, (ii) Updating (iii) Deleting (iv) Search for an element (v) Displaying all elements (in-order).

AVL Trees. Using a C program, perform the following operations on the *AVL tree* data structure: (i) Inserting, (ii) Deleting, (iii) Update, (iv) Searching, (v) Displaying an element

Using Linked List and BST: Construction of a city database using a linked list and binary search tree and the appropriateness of these structures under various demands for the data.

Unit-IV

Sorting and Searching Mechanism: Linear Search, Binary Search, Implementation of Bubble, Insertion, Selection Sort.

Suggested Assignments

Write C programs to perform both linear and binary search on a given random set of integers. The following points should perform by the program

- Take as input an integer, N, which would decide number of integers to be processed and another input an integer, X (0 < X < N+1), which is the key to be searched
- Randomly generate N integers whose values are between 1 to N, multiple entries are allowed
- Output all the indexes (positions) of key in given set of random integers
- Count number of comparisons in the linear and binary searching process, please note comparisons involved in sorting process (in case of binary search) are not to be included
- Output the result in following table:-

Input size (N)	**Number of Comparisons**	
	Linear Search	**Binary Search**
10		
30		
50		
70		
100		

Reference Books:

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.

2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.

3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.

4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill.

5. Kruse, R. L., Leung, B. P., and Tondo, C. L.: Data Structures and Program Design in C; Prentice - Hall of India, 1999.

6. The *C Programming* Language by Brian W.Kernighan , Dennis M. Ritchie

7. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.

8. Schaum‟s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.

9. Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

10. Michael J. Folk and Bill Zoellick, "File Structures" (Second Edition).

11. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition Addison- Wesley, 1997.

12. Schaum‟s Outline Series, "Data Structure", TMH, Special Indian Ed., Seventeenth Reprint, 2009.

13. Mary E. S. Loomes, "Data Management and File Structure", PHI, 2nd Ed., 1989.

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code: CA464/CA466 Title of the Course: Object Oriented Programming (OOP) & Lab

L-T-P: 3-0-3 Credits : 3+1.5

Prerequisite Course / Knowledge (If any): Must have done a programming course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to……

- CO-1: Describe the object-oriented design concepts. (Understand)
- CO-2: Analyse a given computational problem. (Analyze)
- CO-3: Design classes for a given Computational problem (Create)
- CO-4: Apply the UML concepts to model a problem. (Apply)
- CO-4: Create Java programs for the object-oriented design of the given problem. (Create)
- CO-6: Create Java programs which require to use advanced features of Java such as Exception handling, Interfaces, GUI package etc. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3	1	1	1	1	1	1	1	1	1
CO2	1	1	1	2	3	1	1	1	1	1	1	1
CO3	1	1	1	2	1	1	1	1	1	1	1	3
CO4	1	1	3	2	1	1	1	1	1	1	1	1
CO5	1	1	1	1	2	1	1	1	1	1	1	3
CO6	1	1	3	1	2	1	1	1	1	1	1	1

Detailed Syllabus:

UNIT -I: Object Oriented Fundamentals and Modeling: Need for OOP paradigm, What is object orientation and OO Development, Modelling, Abstraction, Three models of OOD, Object and class concepts, Links and Association Concepts, Generalization and Inheritance, N-ary associations, Aggregation, Abstract classes, multiple inheritance, metadata, Reification, Constraints, Derived data, packages.

UNIT -II: Java Basics: History of Java, java data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring java.io.

UNIT -III: State Modelling and Interaction Modelling: Events, states, Transitions and Conditions, State Diagram, Nested state diagram, Concurrency Use-Case model, Sequence model, Activity model, procedural sequence model, Relation between class, state model and interaction model.

UNIT -IV: Hierarchical abstractions, Generalization and Aggregation, Base class object, subclass, subtype, substitutability, forms of inheritance-specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism-method overriding, abstract classes, the Object class Exception handling in Java: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling, Exploring java.util.

UNIT -V: System Design, Class Design and Implementation Modelling: Overview of system design, performance estimation, reuse plan, Subsystems, Management of data storage and global resources, software control strategy and boundary conditions. Overview of class design, Realizing use-cases, designing algorithms, refactoring and design optimization, Overview of implementation, fine tuning of classes, generalization, and realizing associations.

Java Lab Exercises:

1. Basic features of Java such as data types, control structures, loops and arrays (2 Lab Sessions)
2. Working with classes, constructors, methods, objects. (2 Lab Sessions)
3. Using Inheritance, Polymorphism, Interfaces and abstract classes (3 Lab Sessions). Exercises can include a case study depicting OO application design using polymorphism and inheritance. (For example) Developing a Solitaire Application (Chapter 8 of "Introduction to Object Oriented Programming by Timothy Budd")
4. Exception Handling (1 or 2 Lab Session)
5. Exploring Java IO Package (2 or 3 Lab Sessions)
6. Java GUI Programming such as Applets (2 Lab Sessions)
7. Miscellaneous Topics such as generic classes, collection framework and java.util packages (1 or 2 Lab Sessions)

Reference Books:

1. Herbert Schildt, Java: The complete reference, McGraw hill.
2. Paul J. Deitel and Harvey M. Deitel , Java: How to Program, Prentice Hall.
3. T. Budd, Understanding OOP with Java, Pearson Education.
4. Michael Blaha and James Rambaugh, Object Oriented Modelling and Design with UML, 2nd edition, Eastern Economy Edition.
5. Herbert Schildt, Java: A Beginner's Guide, McGraw Hill Education (India) Private Limited.
6. Bruce Eckle, Thinking in Java, Prentice Hall.
7. Joshua Bloch, Effective Java, Createspace Independent Pub.
8. Kathy Siera, Head First Java, O'Reilly Media

Name of the Academic Program: Master of Computer Applications (MCA)

Course Code:CA467 Title of the Course: Internet Technologies

L-T-P: 2-0-2 Credits: 3

Prerequisite Course / Knowledge (If any):Nill

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Design web pages using scripting languages, cascading styles sheets and identify its elements and attributes. (Create)

CO-2: Develop web pages using client-side technologies and perform event handling and validation procedures. (Create)

CO- 3: Create schemas and documents using markup languages, design and develop lightweight data-interchange formats for exchange of data between client and server applications. (Create)

CO-4: Apply JavaScript libraries to create dynamic web pages, access and use web services for interactive web contents. (Apply)

CO-5: Develop applications using server-side technologies, implement session management, database connectivity, and create dynamic HTML content with PHP. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	3	2	1	1	1	1	1	1	1	1
CO2	2	1	3	2	1	1	1	1	1	1	1	1
CO3	2	1	2	3	1	1	1	1	3	1	1	1
CO4	2	1	2	3	1	1	1	1	2	1	1	1
CO5	2	1	1	2	3	1	1	1	1	1	1	1

Detailed Syllabus:

UNIT- I: Introduction to Web Technologies

Client Server Architecture, Local Host,Introduction To HTML5, W3C, web publishing, Introduction to Style sheet, types of style sheets- Inline, External, Embedded CSS, text formatting properties, CSS Box Model, CSS Border, margin properties, Positioning Use of classes in CSS, color properties, use of <div>&<span>, Layout Design using CSS,Bootstrapping.

UNIT- II: JavaScript

Introduction of JavaScript, JavaScript identifiers, operators, control & Looping structure, Introduction of Array, Array with methods, Math, String, Date Objects with methods User defined & Predefined functions, DOM objects, Window Navigator, History, Location, Event handling, Validations on Forms. Introduction to TypeScript (TS), TS simple,special types, TS arrays,tuples,object types,Enums,Union Types,TS Functions,TS Casting, TS classes, generic and Utility types.

UNIT- III: Client Side Technologies

Introduction to XML and JSON, Introduction to jQuery, Syntax Overview, Anatomy of a jQuery Script, Creating first jQuery script, Traversing the DOM, Selecting Elements with jQuery, Refining & Filtering Selections, Selecting Form Elements, Working with Selections, Chaining, Getters & Setters, CSS, Styling, & Dimensions, Manipulating Elements, Getting and Setting Information about Elements, Moving, Copying, and Removing Elements, Creating New Elements, Manipulating Attributes, Utility Methods, Events, Connecting Event to Elements, Namespacing Events, Event handling, Triggering Event handlers, Event Delegation, Animating effects, animate(), click(), hover(), toggle(), Plugins , Create a basic plugin, Finding & Evaluating Plugins, Writing Plugins, Tabs, Panels and Panes examples, jQuery UI and Forms.

UNIT- IV: AJAX and React JS

AJAX Overview, JQuery's AJAX related methods, Ajax and Forms, Ajax Events. React JS framework, Client side programming, server side programming, Database Programming.

UNIT- V: Server-Side Programming

Introduction to PHP Programming Language, Basic Programming Features: Expressions and Control flow and Arrays, Form Handling, Database Management using PHP and MySQL, Stateful Server Programming with Cookies and Sessions, Apache Server, Node JS programming, Use of POSTMAN, HTTP Methods with Restful Architecture.

Reference Books:

1. Internet and World Wide Web How to Program, P.J. Deitel, H.M. Deitel.
2. Jeffrey C. Jackson, Web Technologies - A Computer Science Perspective, Pearson Education, 2006.
3. Complete reference HTML.
4. JavaScript Bible.

5. HTML, DHTML, JavaScript, Perl & CGI Ivan Bayross
6. XML: How to program Deitel & Deitel.
7. Learning PHP, MySQL & Java Script, Robin Nixon, O'Reilly Publications, 2018

Suggested Exercises as per the Theory syllabus order

Unit 1:

#1.HTML5 & CSS Layout Design for a Responsive Web Page

Objective: Design a responsive web page using HTML5 and CSS, adhering to W3C standards.

Task: Create a multi-section webpage for a fictional company. The page should include a header, navigation bar, main content area, sidebar, and footer. Use HTML5 semantic elements like <header>, <nav>, <section>, <article>, and <footer>.

- Apply Inline, External, and Embedded CSS styles in different parts of your webpage.
- Use Bootstrap to make the layout responsive across different screen sizes.
- Add custom styles using CSS Grid or Flexbox to achieve a complex layout without relying solely on Bootstrap.

#2 Styling and Web Publishing

Objective: Combine knowledge of HTML5, CSS, and web publishing to create and publish a personal blog.

Task: Design a blog page using HTML5, implementing all necessary HTML tags, including <div> and <span> for styling. Use an External CSS stylesheet to manage the styles, including text formatting, color properties, and responsive design.

- Incorporate a custom color scheme and font choices using Google Fonts.
- Publish the blog on a free hosting service like GitHub Pages or Netlify, ensuring it adheres to W3C web standards.
- Extend the project by adding a commenting system using a third-party service like Disqus and ensure your blog is SEO-friendly.

#3 CSS Box Model and Advanced Positioning

Objective: Master the CSS Box Model, positioning techniques, and the use of <div> & <span> elements.

Task:Create a webpage that showcases a product gallery. Each product should be displayed within a card that includes an image, title, and description.

- Use the CSS Box Model to control the padding, borders, and margins of the product cards.
- Implement positioning techniques to create a fixed header and footer, with the gallery in a scrollable central section.
- Experiment with different positioning properties (static, relative, absolute, fixed, and sticky) to place elements in desired locations.

- Challenge yourself by creating an overlay effect on product images using position: absolute; inside the cards.

Unit 2:

#4 Dynamic Form Validation with JavaScript and Event Handling

Objective: Create a dynamic form with client-side validation using JavaScript.

Task: Design a registration form with fields such as name, email, password, date of birth, and phone number.

Implement real-time validation using JavaScript to check:

- Name: Only alphabets, no special characters.
- Email: Valid email format.
- Password: At least 8 characters, including uppercase, lowercase, number, and special character.
- Date of Birth: User must be at least 18 years old.
- Phone Number: Only digits and exactly 10 characters long.
- Utilize DOM manipulation to show error messages next to each field when validation fails.
- Handle form submission only if all fields pass validation.
- Add custom validation rules and create a password strength indicator using the Math object.

#5 Building a To-Do List Application with JavaScript Arrays and DOM Manipulation

Objective: Use JavaScript to create an interactive To-Do list application.

Task: Create a simple UI with an input field to add tasks and a list to display them.

Use JavaScript arrays to store and manage the list of tasks. Implement features to add, remove, and mark tasks as completed. Add a filter to display all tasks, only completed tasks, or only pending tasks.

Use DOM manipulation to update the UI based on the user’s actions.

Extend the functionality by adding a "due date" feature, using the Date object, and sorting tasks based on the due date.

#6 Introduction to TypeScript - Refactoring JavaScript Code

Objective: Refactor a JavaScript codebase into TypeScript.

Task: Take an existing JavaScript project (like the To-Do list from Exercise 5) and convert it to TypeScript. Define types for variables, function parameters, and return values.

- Use TypeScript arrays, tuples, and object types to manage data.
- Implement a Task class to encapsulate the properties and methods related to tasks.
- Use Enums for task statuses (e.g., pending, completed).
- Compile the TypeScript code to JavaScript and test its functionality.

#7 Creating a Weather App with TypeScript and DOM Manipulation

Objective: Build a simple weather app using TypeScript that fetches data from a public API.

Task: Use the OpenWeatherMap API to fetch weather data based on the user’s location.

Create a user interface with an input field to enter a city name and a button to fetch weather data.

- Use TypeScript for data handling, ensuring all API responses are correctly typed.
- Display the weather data (temperature, humidity, weather condition) dynamically on the page.
- Use DOM objects and Event Handling to update the UI based on user input.
- Add error handling for network issues and invalid city names, and use TypeScript's Union Types and Casting to handle different possible API responses.

<u>Unit 3:</u>

#8 Converting XML Data to JSON and Manipulating with jQuery

Objective: Work with XML and JSON data formats and manipulate them using jQuery.

Task: Create an XML file representing a catalog of books, including details like title, author, genre, price, and publication date.

- Write a jQuery script to load the XML data using $.ajax(), parse it, and convert it to JSON format.
- Display the JSON data in a styled HTML table, using jQuery to dynamically generate the table structure.
- Allow users to filter the books by genre using a dropdown menu. The filter should dynamically update the displayed table.
- Implement additional filtering options such as author and price range, using jQuery to refine the selection.

#9 Building a Dynamic Form with jQuery and JSON

Objective: Create a form that dynamically changes based on JSON data, using jQuery to manipulate the DOM.

Task: Create a JSON object that represents a form structure (e.g., a registration form with fields like name, email, password, and country).

- Write a jQuery script to dynamically build the form based on the JSON data.
- Use jQuery to add interactivity, such as displaying additional fields when the user selects a specific country (e.g., showing a "State" dropdown when "USA" is selected).
- Validate the form fields using jQuery before submission, displaying error messages next to invalid fields.
- Extend the form to include conditional logic, such as showing different sets of fields based on the user's selections, and use jQuery to handle complex validation scenarios.

#10 Creating a jQuery Plugin for Tabbed Navigation

Objective: Develop a custom jQuery plugin that implements tabbed navigation for a web page.

Task: Design a simple webpage with three content sections, each associated with a tab (e.g., "Home", "About", "Contact").

- Write a jQuery plugin that enables tabbed navigation between these sections. The plugin should handle the showing and hiding of content, and update the active tab's styling.

- Include options in the plugin for customizing the active tab's CSS class, animation speed, and default active tab.
- Use the plugin to initialize tabbed navigation on your webpage, ensuring that the content loads smoothly.
- Add features like URL hash navigation (so users can bookmark specific tabs) and keyboard accessibility (e.g., using arrow keys to navigate tabs).

#11 Advanced Event Handling and Animations with jQuery

Objective: Create a feature-rich interactive webpage using advanced jQuery event handling and animations.

Task: Develop a web page with a list of items (e.g., products). Each item should have a "Details" button that reveals more information about the item when clicked.

- Use jQuery to implement the click event, with smooth slide-down animation to reveal the details.
- Implement event delegation so that new items added dynamically to the list also have the "Details" functionality.
- Add hover effects using jQuery to highlight items as the user moves their mouse over them, and toggle the item's background color when clicked.
- Enhance the page by adding a toggle feature that allows users to switch between list and grid views, with jQuery handling the layout changes and animations.

Unit 4

#12 Building a Full-Stack Application with AJAX and jQuery

Objective: Develop a complete web application that integrates client-side and server-side programming with AJAX.

Task: Create a server-side API using Node.js and Express that manages a simple inventory of products. The API should support CRUD operations (Create, Read, Update, Delete) for the products.

- Develop a front-end interface using HTML, CSS, and jQuery. Use jQuery's AJAX methods to asynchronously fetch, add, update, and delete products from the server-side API.
- Implement a form to add new products, and use AJAX to submit the form data to the server without refreshing the page. Display the updated product list dynamically.
- Handle errors and display appropriate messages to the user for failed operations (e.g., network issues, invalid inputs).
- Add user authentication and authorization using JSON Web Tokens (JWT) and ensure that certain operations (like deleting a product) require the user to be logged in.

#13 React JS with AJAX for Real-Time Data Fetching

Objective: Build a React application that fetches and displays real-time data using AJAX.

Task: Create a React application that displays real-time weather information for a user's location.

- Use the fetch API or Axios to retrieve weather data from a public API like OpenWeatherMap. Implement the data fetching in a React component's lifecycle method (e.g., useEffect).

- Display the weather data in a user-friendly format, including temperature, humidity, and weather conditions.
- Implement a search feature that allows users to input a city name and update the displayed weather data based on their search, using AJAX to fetch the new data.
- Add additional features such as a 5-day weather forecast and implement error handling for invalid city names or API request failures.

#14 Dynamic Form Submission with AJAX and jQuery in a React App

Objective: Integrate jQuery's AJAX methods into a React application for dynamic form submission.

Task: Build a React-based contact form that includes fields for name, email, subject, and message.

- Use jQuery's AJAX methods within the React component to submit the form data to a server-side endpoint (e.g., built with Node.js).
- Upon successful form submission, display a confirmation message without reloading the page. If the submission fails, show an error message.
- Implement form validation on both the client-side (using React state) and server-side (using Express middleware) to ensure all fields are correctly filled out.
- Extend the form to include file upload functionality, allowing users to attach files with their message. Use AJAX to handle the file upload and provide feedback on the upload status.

#15 Full-Stack CRUD Application with React, AJAX, and Database Integration

Objective: Create a full-stack CRUD (Create, Read, Update, Delete) application using React, AJAX, and a database.

Task: Set up a server-side API using Node.js, Express, and MongoDB (or another database) to manage a collection of users.

- Build a React front-end that displays a list of users fetched from the database. Use AJAX calls to interact with the server-side API for retrieving, adding, updating, and deleting users.
- Implement a form to add and edit users, with the form data submitted via AJAX to the API. Update the React state to reflect changes in the user list dynamically.
- Ensure the application supports pagination for the user list, and handle sorting and filtering operations on the client side using React.
- Implement optimistic UI updates, where the React UI updates immediately after an operation is initiated, and then rolls back if the operation fails on the server.

Unit5

#16 Developing a User Authentication System with PHP and MySQL

Objective: Build a complete user authentication system using PHP, MySQL, and stateful session management.

Task: Set up a MySQL database with a users table to store user information, including username, email, password (hashed), and registration date.

- Create a PHP script to handle user registration. Validate the form data, hash the passwords, and store the user information in the database.
- Implement a login system using PHP sessions. When a user logs in, validate their credentials, and if correct, start a session to keep them logged in.
- Build a profile page that displays the logged-in user’s details and allows them to update their information. Use PHP sessions to ensure that only logged-in users can access the profile page.
- Add a "Remember Me" feature using cookies to keep the user logged in across sessions, and ensure security by preventing session hijacking.

#17 Building a Content Management System (CMS) with PHP and MySQL

Objective: Create a simple CMS using PHP for managing articles in a MySQL database.

Task: Set up a MySQL database with a posts table containing fields for id, title, content, author, and date.

- Develop a PHP backend that allows users to create, edit, and delete articles. Use PHP form handling to process the form data and interact with the MySQL database.
- Implement a frontend interface to display the list of articles, with pagination. Allow users to click on an article title to view its full content on a separate page.
- Add user authentication and authorization, so that only authenticated users can create, edit, or delete articles.
- Implement an image upload feature for articles, storing the images on the server and linking them in the article content.

#18: Creating a RESTful API with Node.js and Testing with POSTMAN

Objective: Build a RESTful API using Node.js and test it with POSTMAN.

Task: Set up a Node.js server using Express to create a RESTful API that manages a list of products.

- Implement CRUD operations: Create, Read, Update, Delete for the products, where each product has a name, description, price, and category.
- Connect the API to a MongoDB or MySQL database to persist the product data.
- Use POSTMAN to test all API endpoints, ensuring that each operation works correctly. Document the API using POSTMAN collections.
- Add user authentication to the API using JWT, and ensure that only authenticated users can create, update, or delete products.

#19 Building a RESTful Web Service with Client-Server Architecture

Objective: Create a simple RESTful API that follows the Client-Server Architecture using Node.js and Express.

Task: Develop a RESTful API for a basic task management system. The API should support the following HTTP methods: GET, POST, PUT, and DELETE.

- Implement endpoints to add, update, delete, and retrieve tasks. Each task should include an ID, title, description, and status (e.g., pending, completed).

- Test your API using Postman or a similar tool to ensure all HTTP methods are correctly implemented.

#20 Stateful Server Programming with PHP Sessions and Cookies, Integrated with Apache Server

Objective: Develop a web application that utilizes stateful server programming using PHP sessions and cookies.

Task: Create a PHP application hosted on an Apache server that tracks a user's browsing history within the application using sessions.

- Implement a shopping cart feature where users can add items to their cart, with the cart state maintained across pages using PHP sessions.
- Use cookies to remember user preferences, such as theme settings (dark mode or light mode), and apply these settings every time the user visits the site.
- Develop an admin panel that allows administrators to view user activity (e.g., products viewed, items added to the cart) by analyzing session data.
- Implement session timeout management, where inactive users are automatically logged out after a certain period of inactivity, and use cookies to store the user's preferred language settings across different sessions.

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-III)**

Course Code : CA512 Title of the Course : **Software Engineering**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Explain the models of software development process (Understand)
- CO-2: Evaluate the appropriateness of different models of software development for their application in various domains (Evaluate).
- CO-3: Apply the requirements engineering to software systems. (Apply)
- CO-4: Describe Software Architectures (understand).
- CO-5: Assess the applicability of software architectures for various combinations of non-functional requirements (Evaluate level).
- CO-6: Apply object oriented and structured and structured paradigms to design software systems (Apply).
- CO-7: Apply testing strategy to test software applications (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	1	2	1	1	3	1	1	1	1	1	1
CO2	1	1	1	2	3	1	1	1	1	1	1	1
CO3	1	1	1	1	1	1	1	1	1	2	1	3
CO4	1	2	3	1	1	1	1	1	1	1	1	1
CO5	1	1	1	1	2	1	1	3	1	1	1	1
CO6	1	3	2	1	1	1	1	1	1	1	1	1
CO7	1	3	2	1	1	1	1	1	1	1	1	1

Detailed Syllabus:

UNIT-I: Introduction to Software Engineering

Need of software engineering, systems engineering, challenges in software engineering, Software process models, quality characteristics of software systems, Ethics in Software Engineering.

UNIT-II: Requirements Engineering

Requirements engineering process, requirements specification, structured and object-oriented analysis

UNIT-III: Software Design

Architectural design, detailed design, Structured and object-oriented design, user interface design

UNIT-IV: Software Testing

Verification, Validation, testing techniques, Testing Process

UNIT-V: Tools and Evolution

CASE Tools, Reverse engineering, Reengineering and Configuration management.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition.

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

3. Grady Booch, James Rumbaugh, Ivor Jacobson(2005), "*The Unified Modeling Language User Guide*", 2nd Edition, Addison Wesley Professional.US

School of Computer and Information Sciences

Name of the Academic Program: **Master of Computer Applications (MCA-III)**

Course Code : CA515 Title of the Course: **Software Engineering Lab**

L-T-P : 0-0-3 Credits : 1.5

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate level.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Create user stories (Create).
- CO-2: Develop test plans for test first development (Create).
- CO-3: Design & develop the stories (Create).
- CO-4: Create the documentation (Create).
- CO-5: Develop Software requirements specification document (Create).
- CO-6: Apply object oriented and structured paradigm (Apply).
- CO-7: Generate test reports (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	2	1	1	1	1	1	1	1	3
CO2	3	1	1	2	1	1	1	1	1	1	1	1
CO3	1	1	1	2	1	1	1	1	1	1	1	3
CO4	1	1	1	1	1	1	1	1	1	2	1	3
CO5	1	1	2	3	1	1	1	1	1	1	1	1
CO6	3	1	2	1	1	1	1	1	1	1	1	1
CO7	1	1	3	1	2	1	1	1	1	1	1	1

Detailed Syllabus:

For a given case study/problem statement, the following deliverables are to be realized

- Define stories
- Identify tasks and develop test plan for stories/task (with the help of specifications)
- Design and develop increments
- Test the increments and release the increment
- Apply object oriented and structured modelling
- Implement the case study for plan driven approach by writing use case specification, designing the system and implementing the same.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

Detailed Syllabus:

UNIT-I: ERP Overview: Integrated management information systems, Supply chain management, Integrated data model, Benefits of ERP, Evolution of ERP and Modern enterprise, BPR (Business Process Reengineering) & ERP, Business modelling for ERP

UNIT-II: Customer Service

UNIT-III: Production Planning and execution

UNIT-IV: Purchasing and goods receipt

UNIT-V: Financial and other metrics

UNIT-VI: ERP Packages

UNIT VII: Case studies, Insurance industry, Banking industry, Pharmaceutical industry, Health care, Consumer products, Retail industry, University, Transport Industry, Telecom industry, Public Sector Industry

UNIT VIII:: Current Trends in Implementations; Hardware / Network selections ; Data Management requirements; Integration requirements and techniques; Other Non-functional requirements necessary for implementations

Reference Books:

Learning Resources/Text Books:

1. 3. Infor ERP - LN Workbook
2. 4. Enterprise Resource Planning – Alexis Leon, Tata McGraw Hill

Suggested Reading

3. INTRODUCTION TO MATERIALS MANAGEMENT 6/E (English) 6th Edition Authors: Tony Arnold / Stephen Chapman Publishers: Pearson India
4. Manufacturing Planning and Control for Supply Chain Management (APICS / CPIM Certification Edition) Authors: Thomas E. Vollmann, CFPIM, William L. Berry, D. Clay Whybark, and F. Robert Jacobs Publishers: McGraw Hill Education
5. Accounting Handbook 6th Edition Authors: Shim Siegal Publishers: Barrons Educational Series
6. Operations Strategy 4th Edition Authors: Nigel Slack and Mike Lewis Publishers: Pearson
7. CONCEPTS IN ENTERPRISE RESOURCE PLANNING Authors: Ellen F. Monk, Bret J. Wagner
8. Enterprise Resource Planning – Ashim Raj Singla, Cengage Learning

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-III)

Course Code: CA511	Title of the Course: Algorithms
L-T-P: 3-1-0	Credits : 4

Prerequisite Course / Knowledge (If any): Data Structures in under graduate level, discrete mathematical structures, knowledge of sorting algorithms and basic search strategies

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Assess the inherent structure/hardness of a problem (Evaluate)

CO-2: Select an appropriate strategy to solve a problem (Understand)

CO-3 Design an algorithm that suits the time complexity requirements of the problem. (Create)

CO-4: Estimate the time and space complexities of an algorithm along with the necessary mathematical proofs when necessary. (Evaluate)

CO-5: Devise algorithms by choosing appropriate data structures (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	3	1	1	1	1	1	1	1	1	1
CO2	1	1	2	3	1	1	1	1	1	1	1	1
CO3	3	1	1	2	1	1	1	1	1	1	1	1
CO4	1	2	3	1	1	1	1	1	1	1	1	1
CO5	1	3	2	1	1	1	1	1	1	1	1	1

Detailed Syllabus

UNIT-I: Analysis of Algorithms: Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substitution method, generating functions, Master's theorem etc. Warm-up to complexity analysis: Heap data structure, priority queue application, Best, worst and average case analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms. Strategies for problem solving

UNIT-II: Divide and Conquer strategy: Time complexity analysis for Merge Sort and Quick Sort Algorithms

UNIT-III: Greedy strategy: Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem (Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis

UNIT-IV: Dynamic Programming strategy: Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memorization. Applications to problems like Coin change, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree (OBST), Chained Matrix Multiplication, Traveling Salesperson Problem (TSP) etc.

UNIT-V: Backtracking and Branch & Bound strategies: State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

UNIT-VI: Theory of NP-Completeness: Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Satisfiability (SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Coloring, NP-Completeness proofs.

Reference Books:

1. Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest, PHI, 3rdEdition 2009.
2. Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
3. Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996
4. The Algorithm Design Manual- Steven S. Skiena, Springer, 2009

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA-II)

Course Code : CA 461 Title of the Course : Computer Networks

L-T-P : 3-1-0 Credits : 4

Prerequisite Course / Knowledge (If any): C Programming, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Distinguish between multiplexing techniques (Understand)
- CO-2: Evaluate the different types of switched networks (Analyze)
- CO-3: Explain the functionalities media access for data-link and network protocols. (Understand)
- CO-4: Apply IP addressing and routing algorithms to design networks by subnetting/supernetting (Apply)
- CO-5: Describe the essential principles such as reliable data transfer, flow control, congestion control of a transport layer protocol (Understand)
- CO-6: Predict the topology given the routing protocol messages (Apply)
- CO-7: Analyze and capture network traffic using simulation tools. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	3	1	2	1	1	1	1	1	1	1	1
CO2	1	3	2	1	1	1	1	1	1	1	1	1
CO3	1	1	2	1	3	1	1	1	1	1	1	1
CO4	2	1	1	3	1	1	1	1	1	1	1	1
CO5	1	2	3	1	1	1	1	1	1	1	1	1
CO6	1	3	1	1	2	1	1	1	1	1	1	1
CO7	2	1	1	3	1	1	1	1	1	1	1	1

Detailed Syllabus:

UNIT - I: Physical Layer: Modulation Techniques: Amplitude, Frequency and Phase, ADSL, Multiplexing Techniques: Frequency division multiplexing, Time division multiplexing, wave length division multiplexing, Differential PCM, Switching Techniques: Circuit, message and packet switching.

UNIT - II: Data Link Layer: PPP, PPPoE, MAC Layer: Ethernet (incl. manchester encoding), Switched Ethernet, VLANs, Spanning Tree Protocol.

UNIT - III: Network Layer: Data Plane: Internet Protocol Addressing: CIDR, Internet Protocol Datagram (including fragmentation and reassembly, routing options), IP Forwarding Algorithm, ARP, ICMP (including ICMP Redirect, ICMP Path MTU discovery, ICMP Destination Unreachable options).

UNIT - IV: Transport Layer: UDP, TCP sliding window protocol, TCP connection establishment, TCP reliability including cumulative and delayed acknowledgements, Nagle algorithm, Karn's algorithm for RTT and RTO estimation, TCP AIMD Congestion Control Algorithm, TCP half-close connections including TCP keepalive timer and probe timer, TCP Fast Retransmit and Fast Recovery.

UNIT - V: Network Layer: Control Plane: Distance Vector Algorithm and Routing Information Protocols V1 and V2, Link State Algorithm and Open Shortest Path First Protocol (OSPF).

UNIT - VI: Application Layer: Domain Naming System (DNS) and Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP.

Reference Books:

1. James F. Kurose and Keith W. Ross. Computer Networking: A top-down approach, 6th edition, Pearson Education.
2. Douglas Comer. Computer Networks and Internets Sixth Edition, 2014. ISBN 0133587932/9780133587937, Pearson Education.
3. Douglas Comer. Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 6th edition, 2013. ISBN-10: 0-13-608530-X ISBN-13: 9780136085300, Pearson Education.
4. Kevin R. Fall and W.Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, 2/E, 2012, ISBN-10: 0321336313 ISBN-13: 9780321336316,Pearson Education.
5. Radia Perlman. Interconnections: Bridges, Routers, Switches, and Internetworking Protocols, 2/E, 2000, ISBN-10: 0201634481 ISBN-13: 9780201634488.Pearson Education.

School of Computer and Information Sciences

Name of the Academic Program: Master of Computer Applications (MCA)

Course Code:CA518 Title of the Course: Python Programming Lab

L-T-P: 0-0-3 Credits: 1.5

Prerequisite Course / Knowledge (If any): Proficiency in any one programming language and understanding of data structures

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

CO-1: Develop Python programs using advanced programming concepts. (Create)

CO-2: Process and analyze textual data using Python (Analyze)

CO- 3: Perform data wrangling, cleaning, and manipulation using Pandas. (Apply)

CO-4: Apply NumPy for numerical computations and matrix operations. (Apply)

CO-5: Create visualizations using Matplotlib and Seaborn for data analysis. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	3	2	1	1	1	1	1	1	1	1
CO2	3	3	3	3	2	1	1	2	1	1	1	1
CO3	3	3	3	2	2	1	1	2	1	1	1	1
CO4	3	3	3	2	3	1	1	2	1	1	1	1
CO5	3	2	2	3	3	1	1	2	1	1	1	1

Detailed Syllabus:

UNIT- I: Python Basics and Advanced Concepts

Syntax, variables, data types (strings, lists, tuples, sets, dictionaries), Conditional Statements, Loops, and Functions, Advanced Concepts: List comprehensions, lambda functions, decorators, File I/O Operations (reading/writing files), Exception Handling and Context Managers

UNIT- II: Text Processing in Python

String manipulation and methods, Regular Expressions (using re module) for pattern matching, Text parsing from files (CSV, JSON, XML), Basic Natural Language Processing (NLP) techniques: tokenization, stemming, lemmatization.

UNIT- III: Numerical Computation with NumPy

Introduction to NumPy: Arrays, Indexing, and Slicing, Element-wise operations and broadcasting, Linear algebra with NumPy: matrix operations, Random number generation and statistical simulations, Performance optimization with vectorization

UNIT- IV: Data Science with Pandas

Introduction to Pandas: Series and DataFrames, Data Importing (CSV, Excel, SQL), Cleaning, and Preprocessing, Handling missing values, duplicates, and outliers, Filtering, Sorting, and Grouping Data, Merging and Joining DataFrames, Introduction to Exploratory Data Analysis (EDA)

UNIT- V: Data Visualization and Advanced Plotting

Introduction to Matplotlib: Line plots, bar charts, histograms, and scatter plots, Customizing plots (legends, titles, labels, colors), Subplots and multiple plots in one figure, Seaborn for statistical visualizations (heatmaps, pair plots, violin plots), Interactive and 3D plotting

Reference Books:

1. "Python for Data Analysis" by Wes McKinney, Shroff/O'Reilly, Third Edition
2. "Python Data Science Handbook" by Jake VanderPlas, Shroff/O'Reilly, First Edition