UTKAL UNIVERSITY

Syllabus

for

Master of Computer Applications (MCA)

(2-Year Programme)

Department of Computer Science & Applications
Utkal University
Bhubaneswar (Odisha)

2020-21

UTKAL UNIVERSITY

Syllabus for Masters of Computer Applications (MCA)

(Applicable for Students Taking Admission from the Session 2020-21)

Objective of the Course

The objective of the MCA curriculum is to equip the students with the ability to analyse varieties of real-life problems and develop computer based solutions for effectiveness and efficiency. Keeping in view the requirements of the evolving software industry and also to provide a foundation for higher studies in Computer Science, effort has been made in the choice of subjects to balance between theory and practical aspects of Computer Science. On successful completion of this course a student can find a career in software industries, corporate sectors, or Government Organizations as a technical professional or pursue research in the core areas of Computer Science and Applications.

Eligibility Criteria

The candidate should have passed BCA/ Bachelor Degree in Computer Science Engineering or equivalent Degree OR Passed B.Sc./ B.Com./ B.A. with Mathematics at 10+2 level or at Graduation Level (with additional bridge courses if required). Obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying Examination.

Syllabus for Masters of Computer Applications (MCA)				
PAPER ID	**PAPER TITLE**	**FULL MARK**		**CREDIT**
		Mid-Term	**End-Term**	
FIRST SEMESTER				
MCA-1.1	Mathematical Foundations of Computer Science	30	70	4
MCA-1.2	Data and File Structures	30	70	4
MCA-1.3	Computer System Architecture	30	70	4
MCA-1.4	Theory of Computation	30	70	4
MCA-1.5	Computer Network	30	70	4
MCA-1.6	Data and File Structures Lab		50	2
MCA-1.7	Computer Network Lab		50	2
SECOND SEMESTER				
MCA-2.1	Data Base Systems	30	70	4
MCA-2.2	Algorithms Design and Analysis	30	70	4
MCA-2.3	Operating Systems	30	70	4
MCA-2.4	Artificial Intelligence	30	70	4
MCA-2.5	Cryptography and Network Security	30	70	4
MCA-2.6	Data Base Systems Lab		50	2
MCA-2.7	Operating Systems Lab		50	2

THIRD SEMESTER				
MCA-3.1	Java Programming	30	70	4
MCA-3.2	Compiler Design	30	70	4
MCA-3.3	Data Warehousing and Data Mining	30	70	4
MCA-3.4	Software Engineering	30	70	4
MCA-3.5	**Elective – 1**	30	70	4
MCA-3.6	Java Programming Lab		50	2
MCA-3.7	Software Engineering Lab		50	2
FOURTH SEMESTER				
MCA-4.1	**Elective – 2**	30	70	4
MCA-4.2	**Elective – 3**	30	70	4
MCA-4.3	**Elective – 4**	30	70	4
MCA-4.4	Project Work / Dissertation	300 *		12
	Total	**2400**		**96**

List of Elective Papers #

Elective-1	
MCA-3.5(1)	Computer Graphics & Animation
MCA-3.5(2)	Distributed System
MCA-3.5(3)	Wireless Sensor Networks
MCA-3.5(4)	Machine Learning
MCA-3.5(5)	Combinatorics and Graph Theory
Elective-2	
MCA-4.1(1)	Intrusion Detection System
MCA-4.1(2)	Mobile Computing
MCA-4.1(3)	Unix Internals
MCA-4.1(4)	Optimization Techniques
MCA-4.1(5)	Human Computer Interaction
Elective-3	
MCA-4.2(1)	Block Chain Technology
MCA-4.2(2)	Cloud Computing
MCA-4.2(3)	Simulation and Modeling
MCA-4.2(4)	Data Science using Python
MCA-4.2(5)	Bioinformatics
Elective-4	
MCA-4.3(1)	Internet of Things
MCA-4.3(2)	Text Analysis
MCA-4.3(3)	Digital Image Processing
MCA-4.3(4)	Network Programming
MCA-4.3(5)	Web Design
MCA-4.3(1)	Internet of Things

* **Mark Distribution for Project Work / Dissertation: Report (200), Presentation (50), Viva Voce (50)**

 # A student can opt for only one paper from among the papers mentioned under the respective elective groups

Title of the Paper: **Mathematical Foundations of Computer Science**

Pre-Requisite: None	**Paper Code:** MCA-1.1	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To introduce the concepts of mathematical logic, sets, relations, and functions and perform the operations associated with sets, functions, and relations.
- To relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
- To introduce generating functions and recurrence relations.
- To use Graph Theory for solving problems.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Apply mathematical logic to solve problems.
- Understand sets, relations, functions, and discrete structures.
- Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, and functions.
- Formulate problems and solve recurrence relations.
- Model and solve real-world problems using graphs and trees.

UNIT-I

Logic and Proofs: Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Normal Forms, Proof Methods and Strategy, Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms.

UNIT-II

Basic Structures: Sets, Set Operations, Functions, Recursive Functions, Sequences and Summations. Relations: Relations and their Properties, n-ary Relations and their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Recurrence Relations, Partial Ordering. Boolean.

UNIT-III

Algebra: Boolean Functions, Representing Boolean Functions, Logic Gates, Minimization of Circuits. Algebraic Structures & Coding Theory: The Structure of Algebras, Semi-groups, Monoids and Groups, Homomorphism, Normal Subgroups, and Congruence Relations, Rings, Integral Domains and Fields, Quotient and Product Algebras, Coding Theory. Polynomial Rings and Polynomial Codes.

UNIT-IV

Counting: Basics of Counting, The Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations. Advanced Counting Techniques, Applications of Inclusion-Exclusion, Discrete probability, Conditional probability, Bayes' Theorem.

UNIT-V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Havel-Hakimi Theorem, Representing Graphs and Graph Isomorphism, Connectivity, Cut-Sets, Euler and Hamiltonian Paths, Shortest-Path Problem, Planar Graphs, Graph Coloring, Network Flows.

Reference Books:

1. Kenneth H Rosen, Discrete Mathematics & Its Applications, McGraw-Hill. 7/e.
2. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009
3. C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985
4. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms, John wiley Publication, 1988

Title of the Paper: **Data and File Structures**

Pre-Requisite: None	**Paper Code:** MCA-1.2	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To understand the fundamentals of different data structure.
- To provide the knowledge of basic data structures and their implementations
- To understand importance of data structures in context of writing efficient programs.
- To have a mathematical foundation in analysis of algorithm.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Learn the basic types for data structure, implementation and application.
- Know the strength and weakness of different data structures.
- Use the appropriate data structure in context of solution of given problem.
- Develop programming skills which require solving given problem

UNIT-I

Introduction and Overview: Definitions, Concept of Data Structures, Overview of Data Structures, Implementation of Data Structures, ADT, Arrays: Terminology, One-Dimensional Array, Multi-Dimensional Arrays, Pointer Arrays. Linked Lists: Single Linked List, Circular Linked List, Double Linked List, Circular Double Linked List, Application of Linked Lists, Memory Representation, Boundary Tag System, De-allocation Strategy, Buddy System, Compaction.

UNIT-II

Stacks: Definition, Representation of Stack (Array, Linked List), Operations on Stacks, Applications of Stack (Evaluation of Arithmetic Expressions, Code Generation, Implementation of Recursion, Factorial Calculation, Quick Sort, Tower of Hanoi, Activation Record Management).

Queues: Definition, Representation of Queues (Array, Linked List), Circular Queue, Dequeue, Priority Queue, Application of Queues (Simulation, CPU Scheduling in Multiprogramming Environment, Round Robin Algorithm).

UNIT –III

Binary Trees – Binary Tree Representations – node representation, internal and external nodes, implicit array representation - Operations on binary trees – Binary tree Traversals – Binary search trees- insertion, deletion, find, Types of Binary Trees (Expression Tree, Binary Search Tree, Heap Tree, Threaded Binary Trees, Height Balanced Binary Tree, Weighted Binary Tree, Decision Trees).

UNIT –IV

Data structures for disjoint sets- Red-black trees – insertion and deletion – B-trees – Definition, insertion, deletion – Splay tree, Binomial heaps – operations. Graphs – Representation – Linked representation of Graphs – Graph Traversals.

UNIT –V

Searching and Sorting Techniques: Selection, Bubble, Insertion, Merge, Quick, and Radix sort - Address calculation - Linear search - Binary search. Fundamentals of File Structures and File Organization, File Sorting and Merging.

Reference Books:

1. S. Lipschutz and G.A.V. Pai, “Data Structures”, Tata McGraw-Hill, 2010.
2. D. Samanta, “Classic Data Structures”:, 2/e (PHI).
3. D.S Malik, “Data Structure using C++”, 2/e, Cengage Learning, 2010
4. Adam Drozdek, "Data Structures and algorithm in C++"*, 3/e*, Cengage Learning, 2012.
5. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson.
6. J. P. Tremblay and P. G. Sorenson, "An Introduction to Data Structures with applications", Second Edition, Tata McGraw Hill,1981

Title of the Paper: **Computer System Architecture**

Pre-Requisite: None	**Paper Code:** MCA-1.3	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To understand the structure, function and characteristics of computer systems.
- To understand the design of the various functional units and components of computers.
- To identify the elements of modern instructions sets & their impact on processor design.
- To explain the function of each element of a memory hierarchy in order to identify and compare different methods for computer I/O.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Understand the major architectural styles and their features.
- Identify outline descriptions of real processors and understand in which way their designs fit into the frameworks described in the course.
- Understand the impact of design choices in programming based on architectural design.

UNIT-I

Memory: Internal - External - Memory Organization - Associative - Cache – Virtual memory, Cache Memory: Computer Memory System, Cache Memory Principles, Elements of Cache Design, Pentium-4 Cache Organization, ARM Cache Organization. Internal Memory: Semiconductor Main Memory, Error Correction, Advanced DRAM Organization.

UNIT-II

External Memory: Magnetic Disk, RAID, Solid State Drivers, Optical Memory, Magnetic Tape. Input/ Output: External Devices, I/O Modules, Programmed I/O, Interrupt Driven I/O, Direct Memory Access, I/O Channels and Processors, The External Interface (Thunderbolt & InfinBand), IBM zEnterprise 196 I/O Structure.

CPU: Arithmetic and Logic Unit - Instruction Sets - RISC - CISC - Instruction pipeline - Addressing modes and formats - Register organization - Control Unit Operation - Processor organization.

UNIT-III

Instruction Sets Characteristics & Functions: Machine Instruction Characteristics, Types of Operands, Intel x86 & ARM Data Types, Types of Operations, Inter x86 & ARM Operation Types. Instruction Sets Addressing Modes & Formats: Addressing Modes, x86 & ARM Addressing Modes, Instruction Formats, x86 & ARM Instruction Formats, Assembly Language.

UNIT-IV

Processors: Parallel – Multi-core – Mobile – Embedded – GPU and TPU, Processor Structure & Functions: Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining, The Processor Family, The ARM Processor. Instruction-Level Parallelism & Superscalar Processors: Design Issues, Pentium-4, ARM Cortex-A8.

UNIT-V

Parallel Processing: Multiple Processor Organization, Symmetric Multiprocessors, Cache Coherence & MESI Protocol, Multi-threading & Chip Multiprocessors, Clusters, Non-uniform Memory Access, Vector Computation. Multicore Computers: Hardware Performance Issues, Software Performance Issues, Multicore Organization, Intel x86 Multicore Organization, ARM11 MPCore, IBM zEnterprise 196 Mainframe.

Reference Books:

1. William Stallings: Computer Organization and Architecture. 9/e
2. William Stallings, “Computer Organization and Architecture”, 7 th Edition, 2006, PHI.
3. Hennessy J. and Patterson D., “Computer Architecture – A Quantitative Approach”, 1990, Morgan Kaufmann.
4. M. Morris Mano, “Computer System Architecture”, 3rd Edition, Prentice-Hall of India,2004.
5. M. Morris Mano and Michael D. Ciletti, “Digital Design”, 4 th Edition, 2007, Pearson Education

Title of the Paper: **Theory of Computation**

Pre-Requisite: None	**Paper Code:** MCA-1.4	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To learn the mathematical foundations of computation including automata theory
- To learn the theory of formal languages and grammars; the notions of algorithm, decidability, complexity, and computability.
- To learn about how really computers works and what kind of activities can be computed mechanically within a computer.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Model, compare and analyse different computational models using combinatorial methods.
- Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
- Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.
- Identify limitations of some computational models and possible methods of proving them.
- Have an overview of how the theoretical study in this course is applicable of application like designing the compilers.

UNIT–I

Regular Language and Finite State Machine, Non-Deterministic Finite State Machine, Equivalent Automata, Regular Expression, Properties of regular languages, Pumping Lemma and its application, Minimization of Finite Automata, Decision Properties of regular languages, Linear Grammar.

UNIT–II

Context Free languages and Push Down Machine, Non-Deterministic Push Down Machine, CFG and Derivation Trees, Properties of CFL, Chomsky Normal Form, Pumping Lemma CFL, Decision Properties of CFL.

UNIT–III

Context Sensitive Languages, Turing Machine, Variants of Turing Machines, Non-Deterministic Turing machine, Enumerators, Decidable languages, Decidable problems concerning regular and context free languages. Halting Problem, Post Correspondence Problem.

UNIT–IV

Undecidable Problem, Reducibility, Undecidable problems about Turing Machine, Rice's Theorem and its application, Mapping reducibility, Computable function

UNIT–V

Complexity Theory, NP-completeness, Space Complexity, Savitch's theorem, PSPACE, PSPACE Completeness, The class L and NL, NL completeness, Intractability, Probabilistic algorithms.

Reference Books:

1. Michael Sipser, "INTRODUCTION TO THE THEORY OF COMPUTATION"
2. J.E. Hopcroft, R.Motwani and J.D.Ullman, "INTRODUCTION TO AUTOMATA THEORY, LANGUAGES, AND COMPUTATION"

Title of the Paper: **Computer Network**

Pre-Requisite: None	**Paper Code:** MCA-1.5	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To learn about computer network organization and implementation.
- To obtain a theoretical understanding of data communication and computer networks**.**
- To gain practical experience in installation, monitoring, and troubleshooting of current LAN systems

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Describe how computer networks are organized with the concept of layered approach.
- Describe how signals are used to transfer data between nodes.
- Implement a simple LAN with hubs, bridges and switches.
- Describe how packets in the Internet are delivered.

UNIT-I

Introduction to computer networks, Types of Computer Networks, networking principles, Layering in the Internet, TCP/IP Protocol Suite, OSI Reference Model.

Physical Layer: Transmission Medium- Guided (wired) medium, Unguided (wireless) medium; Transmission impairments, Channel Capacity of noisy and noiseless channels, Data and signal fundamentals, Data Encoding, Modulation and Transmission: Line encoding schemes, Block encoding schemes, Scrambling techniques, Digital to Analog, Analog to Analog, Analog to Digital. Multiplexing- Frequency division multiplexing, Time division multiplexing – Synchronous and Statistical time division multiplexing, Wavelength Division multiplexing.

Networking with telephone networks: Switching- circuit switching, packet switching, message switching;

Network Devices: Hubs and Repeater.

UNIT-II

Data Link Layer: Link Layer Services, Framing, Error detection and Correction Techniques, Multi Access Protocols, Link Layer Addressing, Point to Point Protocol, *LAN Technology:* Ethernet, Token Ring, FDDI, Wireless LANs-IEEE 802.11, ATM; ATM networks - AAL, virtual circuits, control of ATM networks, *WAN Technology:* Frame Relay, SONET/SDH, Handling variable length data, Pulse stuffing. Concept of spreading spectrum, Frequency hopping spread spectrum and Direct sequence spread spectrum, random access, random/slotted ALOHA, splitting algorithms, CSMA-CD, CSMA-CA, MAC throughput analysis, Multiprotocol Label Switching.

Network Devices: Switches, Bridges.

UNIT-III

Network Layer: Introduction, Virtual Circuit and Datagram Networks, Classful IP Addressing, Subnetting, CIDR, Loopback Address, Routing Algorithms (Link State, Distance Vector, Hierarchical), Routing in the Internet (RIP, OSPF, BGP), Broadcast and Multicast Routing Algorithms, Routers, ARP, RARP, DHCP, ICMP, Network Address Translation (NAT), IPv6. **Network Devices:** Routers, Gateways.

UNIT-IV

Transport Layer: Introduction to Transport Layer Services, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Connection Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control, Sockets, control of networks - QoS, window and rate congestion control, open and closed loop flow control, large deviations of a queue and network;

UNIT-V

Application Layer: Web and HTTP, Domain Name Space (DNS), Electronic Mail (SMTP, MIME, IMAP, POP3), File Transfer Protocol, RPC, Cryptography.

Recommended Books:

1. "Data and Computer Communication", by William Stallings, 9th Edition, Pearson Education, 2011
2. "Computer Networks", by Andrew S. Tanenbaum, 4th Edition, Prentice Hall India, 2003
3. "Computer Networks: A Systems Approach", by Larry L. Peterson and Peter S. Davie,4th Edition, Morgan Kauffman Publishers, 2007

Title of the Paper: **Data and File Structures Lab**

Pre-Requisite: None	**Paper Code:** MCA-1.6	**Duration:** 90 Lectures	**Credit:** 2

OBJECTIVES

- To provide the knowledge of basic data structures and their implementations
- To understand importance of data structures in context of writing efficient programs.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Learn the basic types for data structure, implementation and application.
- Know the strength and weakness of different data structures.
- Use the appropriate data structure in context of solution of given problem.
- Develop programming skills which require solving given problems.

List of Experiments:

1. Array implementation of Stack and Queue ADTs
2. Array implementation of List ADT
3. Linked list implementation of List, Stack and Queue ADTs
4. Applications of List, Stack and Queue ADTs
5. Implementation of Binary Trees and operations of Binary Trees
6. Implementation of Binary Search Trees
7. Implementation of AVL Trees
8. Implementation of Heaps using Priority Queues.
9. Graph representation and Traversal algorithms
10. Applications of Graphs
11. Implementation of searching and sorting algorithms
12. Hashing – any two collision techniques

Reference Books:

1. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson publication.
2. D. Samanta, “Classic Data Structures”:, 2/e (PHI).
3. Niklaus Wirth, “Algorithm + Data structure=Program”, Prentice Hall

Title of the Paper: **Computer Network Lab**

Pre-Requisite: MCA-1.5	**Paper Code:** MCA-1.7	**Duration:** 90 Tutorials	**Credit:** 2

OBJECTIVES

- To learn how do computers and terminals actually communicate with each other.
- To understand the parts of a communication network and how they work together.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Understand network layers, structure/format and role of each network layer.
- Able to design and implement various network applications such as data transmission between client and server, file transfer, real-time multimedia transmission.
- Understand the various Routing Protocols/Algorithms and Internetworking.

List of Experiments:

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption.
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.
10. Wireshark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate& Throughput.
    vi. Simulate to Plot Congestion for Different Source/Destination
    vii. Simulate to Determine the Performance with respect to Transmission of Packets

Reference Books:

1. Mike Halsey, Joli Ballew - Windows Networking Troubleshooting- Apress (2017)

Title of the Paper: **Database Systems**

Pre-Requisite: MCA-1.2	**Paper Code:** MCA-2.1	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To learn the fundamental elements of database system.
- To learn the basic concepts of relational database management systems.
- To learn various SQL commands.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify advance database concepts and database models.
- Apply and analyze various terms related to transaction management in centralized and distributed database.
- Produce data modeling and database development process for object–oriented DBMS.
- Analyze and Implement the concept of object- relational database in development of various real time software.

UNIT-I

Introduction to Database and DBMS, Data Models, Relational Data Model, 12 Golden Rules, Structure of Relational DB, Various Constraints, Data Independence, Data abstraction, DB architecture with 3 level of abstraction, Client-Server Architecture, Mappings. **Data Modeling:** E-R Model, Entity and their types, Attribute, Different Types of Attributes, Relationship and their Types, E-R Diagram, Enhanced E-R Model, Specialization, Generalization, Constraints on Specialization& Generalization, Shema Designing using UML, Ontological Concepts.

UNIT-II

Functional Dependencies, Normal Forms: 1NF, 2NF, 3NF, BCNF, Multi-Value dependencies(4NF), Join Dependencies-5NF, PJNF. **File Organization & Indexing:** Organization of records in Files, Sequential, Indexed-Sequential, Hashed, Indexing Structures: Dense Index, Sparse Index, Primary & Secondary Index, Tree Structured Index (B & B+ trees), Hash-based Index, Multidimensional Index, Bitmap Index.

UNIT-III

Introduction to SQL: Data Integrity, Integrity Constraints, Keys, Various Types of Keys, Basic SQL Commands and their categories, how to create Database, Table and View through SQL, SELECT, CREATE, ALTER, DELETE, DROP, TRUNCATE, JOIN, Aggregate Functions, Group by Clause, Order by Clause, Nested SQL Queries. **Relational Algebra:** Unary & Binary Operations, Operation from Set Theory, additional Operations, Examples of RA Queries.
Relational Calculus: Calculus Vs. Algebra, Tuple Relational Calculus, Domain Relational Calculus, Query by Example (QBE).

UNIT-IV

Query Processing & Optimization: Translating SQL Queries into Relational Algebra, Operator Evaluations, Algorithms for relational operators, Algorithm for (Sorting, SELECT, JOIN, PROJECT and Set Operation), Query Optimization, Cost-based and Heuristic Optimization, Semantic Query Optimization. **Transaction Processing:** Transaction and ACID Property, different anomalies of Transaction processing, Serial Schedule & Serializability, Conflict Serializability, View Serializability, Precedence Graph & Serializability testing.

UNIT-V

Concurrent Transaction Processing: Locking, Types of Lock, Two-Phase Locking, different Variants of 2PL, Time-Stamp Ordering protocol, Thomas Write Rule. **Database Recovery:** Crash Recovery, Log-Based Recovery, Shadow Paging, ARIES Recovery algorithm, Database Backup. **Database Security:** Database Security Issues, Discretionary Access Control, Mandatory Access Control, SQL Injection, Flow Control, inference Control & Encryption.

Recommended Books:

1. "Database Systems- The Complete book", by Hector G Molina, Jeffrey D. Ullman and Jennifer Widom, 2nd Edition, Pearson Education India, 2013.
2. "Database System Concepts", by Silberschatz A, Korth H F and Sudarsan S, 6th Edition, McGrawHill, 2013.
3. "Fundamental of Database System", by Elmasri Ramez and Navathe Shamkant, 7/e, Pearson Educ, 2017.
4. "An Introduction to Database Systems", by C. J Date, A. Kanan and S. Swamynathan, 8/e, Pearson Education India, 2006.
5. "Database Management System", by Ramakrishna R. and Gehrke J.,3rd Edition, McGrawHill, 2003.

Title of the Paper: **Algorithms Design and Analysis**

Pre-Requisite: MCA-1.1	**Paper Code:** MCA-2.2	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To have a mathematical foundation in analysis of algorithm
- To expose students to the basic concepts of algorithm design and analyze its complexity.
- To develop algorithms for problems relating to sorting, database queries optimization, graph networks, transport network, etc.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Design algorithms for various applications such as Robotics, Artificial Intelligence, Machine learning, Computer Networks, Parallel computing, etc.
- Use the appropriate data structure in context of solution of given problem.
- Develop programming skills which require solving given problems.

UNIT-I

Analysis and Design of Algorithm (Case study insertion sort and merge sort) Asymptotic Analysis, Divide and Conquer, Recurence Relations, Strassens Matix Multiplication.

UNIT-II

Sorting: Quick sort, heap sort, Counting sort, lower bound for sorting, Randomized quicksort, Order Statistics.

UNIT-III

Amotized Analysis (Aggregate analysis, accounting analysis, Potential analysis), 2-3-4 tree Advanced Data structure: Fibonacci heap, Redblack tree, hashing, data structure on disjoint set, Sciccinet Data Structure.

UNIT-IV

Dynamic Programming: Matrix Chain multiplication, LCS, TSP, Branch and Bound. Greedy Algoithm: MST: Krushkal, Prims, Dijkstra Algorithm, Huffman Coding, Maxflow matching, Computational geometry: Convex Hall,0-1-knaplock, fractional knapsack, Back tracking (4-Queen Prob.)

UNIT-V

Complexity Class: P. PSPACE, NP, NP-Hard, NP Complete, Satisfiability, Cheque, Vertex Cover, Independent set, Exact cover, Graph Coloring, Hamiltonian, Cycle Matching. Approximation Algorithm: Vertex Cove, TSP, Indepentent Set, Sum of subset.

Recommended Books:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein “Introduction to Algorithms”, PHI, 3/e, 2009.
2. Sarabasse & A.V. Gelder Computer Algorithm, “Introduction to Design and Analysis”, Pearson 3/e, 1999
3. E. Horowitz, S. Sahni, & S. Rajasekaran, “Fundamentals of Computer Algorithms”, 2/e, University Press.
4. A.V. Aho, J.E. Hopcroft, & J.D. Ullman, “The Design and Analysis of Computer Algorithm, Pearson.

Title of the Paper: **Operating Systems**

Pre-Requisite: MCA-1.3	**Paper Code:** MCA-2.3	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To understand Operating system structure and services.
- To understand the concept of a Process, memory, storage and I/O management.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify the low-level structure and internal mechanism of operating system.
- Understanding the performance and design trade-offs in complex software systems.
- Describe the main responsibilities of a contemporary operating system (OS).
- List the most fundamental subsystems of an OS and the functions that each subsystem is responsible.
- Recognize and give examples of conflicting goals and compromises necessary in implementing an OS and configuring its run-time parameters

UNIT- I

Operating System, Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special Purpose Systems, Computing Environments, Open-Source Operating Systems. Operating System Services, User Operating System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating System Structure, Virtual Machines, Operating System Debugging, Operating System Generations. System Boot.

UNIT- II

Process: Process Concept, Process Scheduling, Operations on Processes, Inter-Process Communication, Examples of IPC Systems, Communication in Client-Server Systems. Multithreaded Programming: Multithreading Models, Thread Libraries, Threading Issues, Operating-System Examples.

UNIT- III

Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Thread Scheduling. Multiple-Process Scheduling. Synchronization: The Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Monitors, Synchronization Examples, Atomic Transactions.

UNIT- IV

Deadlocks: System Model, Deadlock Characterization, Methods of Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, Recovery from Deadlock. Memory Management Strategies: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Example: The Intel Pentium.

UNIT- V

Virtual-Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory. File System: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection.

Recommended Books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8/e, John Wiley Pub. 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3/e, Pearson Education 2007
3. W. Stallings, "Operating Systems, Internals & Design Principles", 5/e, Prentice Hall of India. 2008
4. G. Nutt, "Operating Systems: A Modern Perspective", 2/e, Pearson Education 1997.

Title of the Paper: **Artificial Intelligence**

Pre-Requisite: MCA-1.2, MCA-2.2	**Paper Code:** MCA-2.4	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To learn the basic concepts of AI principles and approaches.
- To develop the basic understanding of the building blocks of AI.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Have fundamental understanding of the basic concepts of artificial intelligence (AI).
- Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
- Have fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- Have knowledge of current scope and limitations, and societal implications of AI.
- Have basic foundation of machine learning.

UNIT-I

Introduction to Artificial Intelligence: The Imitation Game and Turing Test, Lady Lovelace's Objection, Learning Machines, the Evolution of AI, Water-Jug problem, Missionaries-Cannibals Problem, Monkey-Bananas Problem, Tower of Hanoi, Tower of Brahma, Cryptarithmetic problem, Production System, Working Memory, Inference Engine.

State Space Search Algorithms: Depth First Search, Breadth First Search, Iterative Deepening, Uniform Cost Search, Hill-Climbing, Beam Search, Tabu Search. *Heuristic Search:* Greedy Best-First Search, A* algorithm, 8-Puzzle Problem, Iterative Deepening A*(IDA*) algorithm, Problem Decomposition, And-Or Graph, AO* algorithm, Rete Net, Rete algorithm, Simulated Annealing, Genetic Algorithms.

UNIT-II

Game Playing Strategies, Game Tree, MIN-MAX, Tic-Tac-Toe, Alpha-Beta Pruning, SSS* algorithm, B* Search. Constraint Satisfaction Problems, N-Queen problem.

Knowledge Representation: Types of Knowledge, presentation by Logic, propositional logic, substitutional rules, *Rules of Inference:* Modus Ponens, Modus Tollens, Resolution in Propositional Logic, Refutation, Predicate Logic, First Order Logic Representation, Skolemization, Natural Deduction, Resolution in Predicate Logic, Clause Form, Prenex Normal Form, Unification.

UNIT-III

Knowledge Representation using Rules: PROLOG, Facts, Rules, Defining Relations by rules and facts, Recursive rules, Atoms, Variables, Data & Structured Objects, Lists: Membership, Concatenation, adding to & Deleting from List, sublist, permutations, Operators in PROLOG, Cuts, problems with Cuts, Managing Input & Output in PROLOG, Control Facilities, Database Manipulation, bagof, setoff, and findall. Forward Reasoning, Backward Reasoning, Matching, Statistical Reasoning, Bayesian Belief Network, Dempster-Shafer Theory, Fuzzy Logic.

UNIT-IV

Planning: Planning Agent, PDDL, STRIPS approach, TWEAK approach, Blocks World Problem.

Machine learning paradigms: Introduction, machine learning systems, supervised and unsupervised learnings, inductive learning, deductive learning, clustering, support vector machines, Decision Tree, case-based reasoning and learning, Artificial neural networks: Introduction, artificial networks, single layer feed forward networks, multi layered forward networks, design issues of artificial neural networks.

UNIT-V

Natural Language Processing: Steps in NLP, Parsing, Augmented Transition Network, Semantic Analysis, Discourse and Pragmatic Processing.

Word Level Analysis: Unsmoothed N-grams, Evaluating N-grams, Skip-Gram, Word2vec, Smoothing, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

Spelling Checking: Spelling Errors, Spell checking Techniques, Soundex Algorithm.

Recommended Books:

1. "A First Course in Artificial Intelligence", by Deepak Khemani, 1st Edition, McGrawHill, 2013.
2. "Artificial Intelligence", by Rich E, Knight K and Nair S. B., 3rd Edition, McGrawHill, 2013.
3. "Artificial Intelligence: Structure and Strategies for Complex Problem solving", by George F Luger, 4th Edition, Pearson Education, 2004.
4. "Artificial Intelligence: A Modern Approch", by Russell S. J., Norvig P., 3rd Edition, Pearson, 2015.
5. "PROLOG programming for artificial intelligence", by Ivan Bratko,3rd Edition, Addison-Wesley, 2001.
6. "Programming in PROLOG", by Clocksin W.F and Mellish C.S, Springer-Verlag, 5th Edition,2003.

Title of the Paper: **Cryptography and Network Security**

Pre-Requisite: MCA-1.5	**Paper Code:** MCA-2.5	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- Learn fundamentals of cryptography and its application to network security.
- Understand network security threats, security services, and countermeasures.
- Acquire background on well-known network security protocols such.
- Understand vulnerability analysis of network security.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Understand various Cryptographic Techniques
- Apply various public key cryptography techniques
- Implement Hashing and Digital Signature techniques
- Implement system level security applications

UNIT-I

Introduction: Security Goals, Cryptographic Attacks, Services and Mechanism, Techniques; Mathematics of Cryptography: Integer Arithmetic, Modular Arithmetic, Matrices; Symmetric-Key Ciphers: Substitution Ciphers, Transposition Ciphers, Stream and Block Ciphers.

UNIT-II

Groups, Rings, Field, GF(2^n) Fields; Data Encryption Standard (DES); Advanced Encryption Standard (AES); Euler's Phi-Function, Chinese Remainder Theorem (CRT), Exponentiation and Logarithm.

UNIT-III

Asymmetric-Key Cryptography: RSA Cryptosystem; ElGamal Cryptosystem; Hash Function: MD, SHA; Digital Signature; Digital Signature Schemes: Digital Signature Standard (DSS).

UNIT-IV

Security at the Application Layer: E-mail, PGP, S/MIME; Security at the Transport Layer: Secure Socket Layer (SSL) Architecture, SSL Message Formats, Transport Layer Security (TLS).

UNIT-V

Security at the Network Layer: Two Modes, Two Security Protocols, Security Association, Security Policy; System Security: Users, Trust and Trusted Systems, Buffer Overflow and Malicious Software, Malicious Programs, Worms, Viruses, Intrusion Detection Systems (IDS), Firewalls.

Recommended Books:

1. Behrouz A. Frouzan: Cryptography and Network Security, **McGraw-Hill.**
2. William Stallings, "Cryptography and Network Security: Principals and Practice", Pearson Education.

Title of the Paper: **Database Systems Lab**

Pre-Requisite: MCA-2.1	**Paper Code:** MCA-2.6	**Duration:** 90 Tutorials	**Credit:** 2

OBJECTIVES

To make students aware about the elements of SQL and make them capable to designing the robust queries for exact data extraction that the client wants to perceive.

LEARNING OUTCOMES

The Students will understand the practical implementation of database and able to create new database and manage them with SQL.

List of Experiments:

1. Creation of a database and writing SQL queries to retrieve information from the database.
2. Performing Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
3. Creation of Views, Synonyms, Sequence, Indexes, Save point.
4. Creating an Employee database to set various constraints.
5. Creating relationship between the databases.
6. Study of PL/SQL block.
7. Write a PL/SQL block to satisfy some conditions by accepting input from the user.
8. Write a PL/SQL block that handles all types of exceptions.
9. Creation of Procedures.
10. Creation of database triggers and functions
11. Mini project (Application Development using Oracle/ Mysql)
    i. Inventory Control System.
    ii. Material Requirement Processing.
    iii. Hospital Management System.
    iv. Railway Reservation System.
    v. Personal Information System.
    vi. Web Based User Identification System.
    vii. Timetable Management System.
    viii. Hotel Management System

Recommended Books:

1. Mary Beth Roeser, “Oracle Database SQL Language Reference”, 12c Release ORACLE Inc.

Title of the Paper: **Operating Systems Lab**

Pre-Requisite: MCA-2.3	**Paper Code:** MCA-2.7	**Duration:** 90 Tutorials	**Credit:** 2

OBJECTIVES

To make students aware about the UNIX operating systems and about the different Kernels of the Operating Systems.

LEARNING OUTCOMES

The Students will learn how to manage the UNIX operating system and able to get fundamental idea how the operating system works.

List of Experiments:

1. Basics of UNIX commands
2. Write programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir
3. Write C programs to simulate UNIX commands like cp, ls, grep, etc.
4. Shell Programming
5. Write C programs to implement the various CPU Scheduling Algorithms
6. Implementation of Semaphores
7. Implementation of Shared memory and IPC
8. Bankers Algorithm for Deadlock Avoidance
9. Implementation of Deadlock Detection Algorithm
10. Write C program to implement Threading and Synchronization Applications
11. Implementation of the following Memory Allocation Methods for fixed partition
    i. First Fit
    ii. Worst Fit
    iii. Best Fit
12. Implementation of Paging Technique of Memory Management
13. Implementation of the following Page Replacement Algorithms
    i. FIFO
    ii. LRU
    iii. LFU
14. Implementation of the various File Organization Techniques
15. Implementation of the following File Allocation Strategies
    i. Sequential
    ii. Indexed
    iii. Linked

Recommended Books:

1. Sumitabha Das, " Your UNIX: The Ultimate Guide", 5/e, TMH

Title of the Paper: **Java Programming**

Pre-Requisite: MCA-1.1	**Paper Code:** MCA-3.1	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

To Introduce the Paradigm of Object-Oriented Programming and to design efficient programs to solve different types of problems.

LEARNING OUTCOMES

The students will be able to solve the mathematical problems of the paper-1.1 with an object-oriented context of programming and also students will be able to design GUI and web-enabled APPLETS.

UNIT-I

Introduction to Java**:** Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).

UNIT-II

Arrays, Strings and I/O**:** Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files. Object-Oriented Programming Overview**:** Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

UNIT-III

Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata**:** Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

UNIT-IV

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

UNIT-V

Applets and Event Handling: Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, text fields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Recommended Books:

1. E. Balagurusamy, “Programming with Java”, 4/e, TMH
2. Bruce Eckel, "Thinking Java", 8/e, Pearson India, 2010.
3. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2/e, 2004.
4. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 ,9/e,Printice Hall.2012.

Title of the Paper: **Compiler Design**

Pre-Requisite: MCA-1.4	**Paper Code:** MCA-3.2	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To provide a thorough understanding of the internals of Compiler Design.
- To explore the principles, algorithms, and data structures involved in the design and construction of compilers.
- Topics include context-free grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation, and code optimization.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Realize basics of compiler design and apply for real time applications.
- Introduce different translation languages
- Understand the importance of code optimization
- Know about compiler generation tools and techniques

UNIT-I

INTRODUCTION TO COMPILERS: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator. PARSING: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT-II

BOTTOM UP PARSING: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

UNIT-III

SYNTAX DIRECTED TRANSLATION: Syntax directed definition, construction of syntax trees, S-attributed and L-attributed definitions, translation schemes, emitting a translation. INTERMEDIATE CODE GENERATION: intermediate forms of source programs– abstract syntax tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT-IV

TYPE CHECKING: Definition of type checking, type expressions, type systems, static and dynamic checking of types, specification of a simple type checker, equivalence of type expressions, type conversions, overloading of functions and operators. RUN TIME ENVIRONMENTS: Source language issues, Storage organization, storage-allocation strategies, access to non-local names, parameter passing, symbol tables and language facilities for dynamic storage allocation.

UNIT-V

CODE OPTIMIZATION: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis. CODE GENERATION: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Recommended Books:

1. A.V Aho, M.S Lam, Ravi Sethi, and J.D Ullman, “Compilers, Techniques and Tools”, 2nd Edition, Pearson Publication.

Title of the Paper: **Data Warehousing and Data Mining**

Pre-Requisite: MCA-2.1	**Paper Code:** MCA-3.3	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To introduce students to the basic concepts and techniques of Data Mining
- To develop skills of using recent data mining software for solving practical problems.
- To study the methodology of engineering legacy databases for data warehousing and data mining to derive business rules for decision support systems
- Develop and apply critical thinking, problem-solving, and decision-making skills which can initiate students about research oriented thinking.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Understand what data mining is all about.
- Perform the data preparation tasks and understand the implications.
- Demonstrate an understanding of the alternative knowledge representations such as rules, decision trees, decision tables, and Bayesian networks.
- Demonstrate an understanding of the basic machine learning algorithmic methods that support knowledge discovery.
- Identify alternative data mining implementations and what might be most appropriate for a given data mining task.

UNIT–I

Data preprocessing, Data cleaning, data integration, reduction, data transformation and data discretization. Data warehouse modeling, Data warehouse Design and usage, data warehouse implementation, from data warehousing to data mining.

UNIT–II

Data cube and OLAP, Data cube computation method, Processing Advanced kinds of Queries by exploring cube technology, Multidimensional Data analysis in cube space.

UNIT–III

Mining Frequent Patterns, Associations and correlations. Frequent Itemset mining methods, Apriori algorithms and Pattern growth approach for mining frequent itemset. Pattern Evaluation methods.

UNIT–IV

Classification basic concepts, Decision tree induction, Bayes classification methods, Rule based Classification, Model evaluation and selection, Techniques to improve classification accuracy, Regression techniques, linear, nonlinear regression, logistic regression

UNIT–V

Cluster Analysis basic concepts and methods, Cluster analysis, Partitioning methods, Hierarchical methods, Density-Based methods, Grid-Based methods, Evalution of Clustering.

Recommended Book:

1. J. Han, M. Kamber and J, Pei., "DATA MINING CONCEPTS and TECHNIQUES", 3/ed(Morgan Kaufmann Pub.)
2. Data mining- the text book by Charu c. Agarwal, 3rd edition, Springer
3. Introduction to Data Mining by Michael Steinbach, Pang-Ning Tan, and Vipin Kumar, Pearson, 2016 edition

Title of the Paper: **Software Engineering**

Pre-Requisite: MCA-2.3	**Paper Code:** MCA-3.4	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To learn the way of developing software with high quality and the relevant techniques.
- To introduce software engineering principles for industry standard.
- To focus on Project management domain and Software risks management.

LEARNING OUTCOMES

Upon Completing the Course, Students will be able to:

- Identify, formulate, and solve complex problems by applying different principles of software engineering.
- Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
- Communicate effectively with a range of audiences and recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.
- Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

UNIT-I

Professional Software Development, Software Engineering Ethics, Software Processes, Software Process Models, Process Activities, Coping with Change, The Rational Unified Process, Agile Software Development, Agile Methods, Plan-Driven and Agile Development, Extreme Programming, Agile Project Management, Scaling Agile Methods.

UNIT-II

Requirements Engineering, Functional and Non-Functional Requirements, The Software Requirements Document, Requirements Specification, Requirements Engineering Processes, Requirements Elicitation and Analysis, Requirements Validation, Requirements Management, System Modelling, Context Models, Interaction Models, Structural Models, Behavioural Models, Model-Driven, Engineering, Architectural Design, Architectural Design Decisions, Architectural Views, Architectural Patterns, Application Architectures.

UNIT-III

Design and Implementation: Object-Oriented Design using the UML, Design Patterns, Implementation Issues, Open-Source Development, Software Testing: Development Testing, Test-Driven Development, Release Testing, User Testing, Software Evolution: Evolution Processes, Program Evolution Dynamics, Software Maintenance, Legacy System Management, Dependability and Security.

UNIT-IV

Socio-technical Systems: Complex Systems, Systems Engineering, System Procurement, System Development, System Operation. Dependability and Security: Dependability Properties, Availability and Reliability, Safety, Security. Dependability and Security Specification: Risk-Driven Requirements, Specification, Safety Specification, Reliability Specification, Security, Specification, Formal Specification.

UNIT-V

Dependability Engineering: Redundancy and Diversity, Dependable Processes, Dependable Systems Architectures, Dependable Programming. Security Engineering: Security Risk Management, Design for Security, System Survivability. Dependability and Security Assurance: Static Analysis, Reliability Testing, Security Testing, Process Assurance, Safety and Dependability Cases.

Recommended Books:

1. I. Sommerville, “Software Engineering”, 9/e, Addison Wesley.
2. R. Mall, “Fundamentals of Software Engineering”, 3/e, PHI
3. R.S. Pressman, “Software Engineering”, A Practitioner‘s Approach, 7/e, McGraw-Hill, 2009
4. K.K. Aggarwal and Y. Singh, “Software Engineering”,2/e, New Age International Publishers, 2008

Title of the Paper: **Wireless Sensor Networks**

Pre-Requisite: MCA-1.5	**Paper Code:** MCA-3.5(3)	**Duration:** 45 Lectures	**Credit:** 4

OBJECTIVES

- To understand the basics of Ad-hoc & Sensor Networks.
- To learn various fundamental and emerging protocols of all layers.
- To study about the issues pertaining to major obstacles in establishment and efficient management of Ad-hoc and sensor networks.
- To understand the nature and applications of Ad-hoc and sensor networks.
- To understand various security practices and protocols of Ad-hoc and Sensor Networks.

LEARNING OUTCOMES

After completing the course, students will be able to:

- Explain the basic concepts of wireless sensor networks, sensing, computing and communication tasks.
- Describe and explain radio standards and communication protocols adopted in wireless sensor networks.
- Describe and explain the hardware, software and communication for wireless sensor network nodes.
- Explain the architectures, features, and performance for wireless sensor network systems and platforms.
- Describe and analyze the specific requirements of applications in wireless sensor networks for energy efficiency, computing, storage and transmission.

UNIT–I

Introduction: Fundamentals of wireless communication technology, the electromagnetic spectrum radio propagation, characteristics of wireless channels, modulation techniques, multiple access techniques, wireless LANs, PANs, WANs, and MANs, Wireless Internet.

UNIT–II

Introduction to adhoc/sensor networks: Key definitions of adhoc/ sensor networks, unique constraints and challenges, advantages of ad-hoc/sensor network, driving applications, issues in adhoc wireless networks, issues in design of sensor network, sensor network architecture, data dissemination and gathering.

UNIT–III

MAC Protocols: Issues in designing MAC protocols for adhoc wireless networks, design goals, classification of MAC protocols, MAC protocols for sensor network, location discovery, quality, other issues, S-MAC, IEEE 802.15.4.

UNIT–IV

Routing Protocols: Issues in designing a routing protocol, classification of routing protocols, table-driven, on-demand, hybrid, flooding, hierarchical, and power aware routing protocols.

UNIT–V

QoS and Energy Management: Issues and Challenges in providing QoS, classifications, MAC, network layer solutions, QoS frameworks, need for energy management, classification, battery, transmission power, and system power management schemes.

Recommended Books:

1. F. Zhao and L. Guibas, Wireless Sensor Network: Information Processing Approach, Elsevier, 2009
2. E. H. Callaway, Jr. E. H. Callaway, Wireless Sensor Networks Architecture and Protocols, CRC Press, 2009
3. C. Siva Ram Murthy, and B. S. Manoj, "AdHoc Wireless networks ", Pearson Education - 2008.

Title of the Paper: **Java Programming Lab**

Pre-Requisite: MCA-3.1	**Paper Code:** MCA-3.6	**Duration:** 90 Tutorials	**Credit:** 2

OBJECTIVES

To make the students aware about the basic concepts and techniques of object oriented programming paradigm and make them able to design practical problem solving programs using different elements of java.

LEARNING OUTCOMES

The Students will be able to do the practical implementation of algorithm and different mathematical problems, which will enable them to design java programs in real world scenario.

LIST OF EXPERIMENTS:

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. Develop an applet in Java that displays a simple message.
4. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.
5. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
6. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
7. Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after deletion.
8. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.
9. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
10. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
11. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
12. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
13. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.
14. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
15. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order
16. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Recommended Books:

1. John R. Hubbard, “Programming with JAVA”, McGrawHill, Indian Edition
2. Joe Wigglesworth and Paula McMillan, “Java Programming: Advanced Topics”, CENGAGE
3. Cay Horstmann, “BIG JAVA”, Wiley, 4th Edition

Title of the Paper: **Software Engineering Lab**

Pre-Requisite: None	**Paper Code:** MCA-3.7	**Duration:** 90 Tutorials	**Credit:** 2

OBJECTIVES

To make students aware about the different tools of diagram designing and about the Software Testing tools.

LEARNING OUTCOMES

The Students will be able to design the different Architecture of Software Models with the help of diagrams used in System Analysis and Designing.

List Of Experiment:

Do the following exercises for any two projects given in the list of sample projects or any other projects:

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

Recommended Book:

1. Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, Tata McGraw Hill Int. 7/e, 2009.
2. Ian Somerville, “Software Engineering”, Pearson Education, 8th Edition, 2008.
3. H. Srimathi, H. Sriram, A. Krishnamoorthy, “Object-Oriented Analysis & Design using UML”, SCITECH Pub.