BODOLAND UNIVERSITY

Curriculum Structures for 2 Years Syllabus for MCA
(Master of Computer Application)

Year 2024

PROGRAMME STRUCTURE AND DETAILED CURRICULUM

Bodoland University

DEBARGAON, KOKRAJHAR (B.T.R.)

Curriculum Structures for NEP 2 Years Syllabus for MCA
No. of Papers= 20, Total Credits= 80, Total Marks =2000

SEMESTER-I						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Internal**	**External**	**Total Marks**
MCAADL14014	Programming using C and Data Structure	4	4+0+0	30	70	100
MCAADL14024	Advance Operating System	4	2+0+0	30	70	100
MCAADL14034	Mathematical Foundation of Computer Science	4	4+0+0	30	70	100
MCAADL14044	Digital System and Computer Organization	4	4+0+0	30	70	100
MCAADL14054	Lab I: C and Data structure	4	0+0+4	30	70	100
Total		**20**				500

SEMESTER-II						
Paper code	**Subject Name**	**Credit**	**L+T+P**	**Internal**	**External**	**Total Marks**
MCAADL15064	Database Management System	4	3+0+1	30	50(Theory) +20(Practical)	100
MCAADL15073	Computer Networks & Cyber Security	4	3+0+0	30	70	100
MCAADL15083	Object Oriented Programming in C++ & Java	4	3+0+1	30	50(Theory) +20(Practical)	100
MCAADL15093	Data Mining and Warehousing	4	3+0+1	30	50(Theory) +20(Practical)	100
MCAADL150103	Formal Language and Automata Theory	4	3+1+0	30	70	100
Total		**20**				500

In MCA 3rd Semester, learners will choose either Group-A or Group-B or Group C

Option A- only Coursework

SEMESTER-III							
Paper code	**Subject Name**		**Credit**	**L+T+P**	**Internal**	**External**	**Total Marks**
MCAADL25014	Distributed System		4	4+0+0	30	70	100
MCASPL25024 A/B/C (Students can choose any one paper)	A	Machine Learning	4	4+0+0	30	70	100
	B	Bioinformatics					
	C	Pattern Recognition					
MCASPL25034 A/B/C (Students can choose any one paper)	A	Data Science	4	4+0+0	30	70	100
	B	Cryptography and Network Security					
	C	Big data					
MCASPL25044 A/B/C (Students can choose any one paper)	A	Natural Language Processing	4	4+0+0	30	70	100
	B	Internet of Things					
	C	Block Chain					
MCASPL25054 A/B/C (Students can choose any one paper)	A	Cloud Computing	4	4+0+0	30	70	100
	B	Software testing					
	C	System Software					
			20				500

SEMESTER-IV							
Paper code	**Subject Name**		**Credit**	**L+T+P**	**Internal**	**External**	**Total Marks**
MCASPL25064 A/B (Students can choose any one paper)	A	Graph Theory and Algorithm	4	4+0+0			100
	B	Speech processing					
MCASPL25074 A/B(Students can choose any one paper)	A	Mobile Computing	4	4+0+0	30	70	100
	B	Advance database system					
MCASPL25084: Seminar	SEMINAR		4	0+4+0	30	70	100
MCADIS25098	PROJECT		4+4			100	100
						100	100
Total			20				500

Guidelines for Seminar:

1. A seminar should be given by an individual students, based on topics chosen from the emerging area and technology of Computer Science and Computer Applications.
2. References from journal such as IEEE, ACM etc… shall be used.
3. A report on this seminar with15-20 pages should be prepared.

Option B- only Research

Research thesis/project with minimum 2 conferences papers. Peer reviewed research publication should be encouraged

Option C- Coursework+Research

SEMESTER-III							
Paper code	**Subject Name**		**Credit**	**L+T+P**	**Internal**	**External**	**Total Marks**
MCAADL25014	Distributed System		4	4+0+0	30	70	100
MCASPL25014 A/B/C (Students can choose any one paper)	A	Machine Learning	4	4+0+0	30	70	100
	B	Bioinformatics					
	C	Pattern Recognition					
MCASPL25024 A/B/C (Students can choose any one paper)	A	Data Science	4	4+0+0	30	70	100
	B	Cryptography and Network Security					
	C	Big data					
MCASPL25034 A/B/C (Students can choose any one paper)	A	Natural Language Processing	4	4+0+0	30	70	100
	B	Internet of Things					
	C	Block Chain					
MCASPL25044 A/B/C (Students can choose any one paper)	A	Cloud Computing	4	4+0+0	30	70	100
	B	Software testing					
	C						
			20				

SEMESTER-IV

Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

PROGRAMME LEARNING OUTCOME

MASTERS OF COMPUTER APPLICATION(MCA)

After completion of the MCA Program, learners would be able to:

1. **Computational Knowledge:** Utilize understanding of basic computing principles, specialized computing knowledge, mathematics, and relevant domain expertise to abstract and conceptualize computing models based on specified problems and requirement.
2. **Problem Analysis :**Identify, formulate, review literature, and resolve complex computing issues by drawing well-supported conclusions using foundational principles of mathematics, computer science, and relevant domain fields.
3. **Design /Development of Solutions:** Create and assess solutions for intricate computing challenges, and develop and appraise systems, components, or processes that fulfill defined requirements, considering public health and safety, as well as cultural, social, and environmental factors.
4. **Conduct Investigations of Complex Computing Problems:** Utilize research-based expertise and methodologies, including experimental design, data analysis, and interpretation, and information synthesis to draw valid conclusions.
5. **Modern Tool Usage:** Develop, choose, modify, and use suitable techniques, resources, and contemporary computing tools for complex computing tasks, while recognizing their limitations.
6. **Professional Ethics:** Comprehend and adhere to professional ethics, cyber regulations, responsibilities, and standards of professional computing practice.
7. **Life-long Learning:** Acknowledge the necessity and possess the ability to engage in self-directed learning for ongoing growth as a computing professional.
8. **Project management and finance:** Exhibit knowledge and comprehension of computing and management principles, applying them to one's work as both a team member and leader, to manage projects and operate in multidisciplinary settings.
9. **Communication Efficacy:** Effectively communicate with the computing community and society at large about complex computing tasks by being able to understand and write clear reports, create detailed documentation, deliver impactful presentations, and give and follow precise instructions.
10. **Societal and Environmental Concern:** Comprehend and evaluate societal, environmental, health, safety, legal, and cultural concerns within both local and global contexts, along with the associated responsibilities pertinent to professional computing practice.
11. **Individual and Teamwork:** Operate proficiently both independently and as part of diverse teams, assuming roles as either a member or a leader, within varied multidisciplinary environments.
12. **Innovation and Entrepreneurship:** Recognize a current opportunity and utilize innovation to seize that opportunity, generating value and prosperity for both the individual and society as a whole.

DETAILED SYLLABUS
SEMESTER-I
MCAADL14014: PROGRAMMING USING C AND DATA STUCTURE
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective:

- To provide complete knowledge of C language.
- Students will be able to develop logics which will help them to create programs, applications in C.
- By learning the basic programming constructs they can easily switch over to any other language in future.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Analyze, design and build logical solutions for problems.
- Use appropriate programming constructs to realize computational algorithms.
- Choose appropriate data structures to represent data items in the real world.
- Manage 1/0 operations through C and Python program.
- Apply code reusability with functions and pointers.

Unit-1: C Programming Fundamentals

Data Types, Variables, Operations, Expressions and Statements, Conditional Statements, Functions, Recursive Functions, Arrays, Single and Multi-Dimensional Arrays.

Unit-2: C Programming, Advanced Features

Structures, Union, Enumerated Data Types, Pointers: Pointers to Variables, Arrays and Functions, File Handling, Preprocessor Directives.

Unit-3: Linear Data Structures, Sorting and Searching techniques

Abstract Data Types (ADTs) , List ADT, Array-Based Implementation, Linked List, Doubly-Linked Lists, Circular Linked List, Stack ADT, Implementation of Stack, Applications, Queue ADT, Priority Queues, Queue Implementation, Applications, nsertion Sort, Quick Sort, Heap Sort, Merge Sort -Linear Search, Binary Search.

Unit-4: Non-Linear Data Structures

Trees, Binary Trees, Tree Traversals, Expression Trees, Binary Search Tree, Hashing, Hash Functions, Separate Chaining, Open Addressing, Linear Probing- Quadratic Probing, Double Hashing, Rehashing.

SUGGESTED READING:

1. Y. Kanetkar (2004). Let us C, BPB Publication.
2. E. Balaguruswami (2004). Programming in ANSI C 2nd Edition, Tata Mcgraw Hill.
3. S.K. Srivastava and Deepali Srivastva. Data Structures through C in Depth, BPB publication.
4. E. Horowitz & S. Sahani (2008), Fundamentals of Algorithms, University Press.
5. AM Tanenbaum, Y Langsam and MJ Augustein (2019), Data structure using C Prentice Hall India.
6. S Lipschutz (2006), Data Structures, TMH.

MCAADL14024: ADVANCE OPERATING SYSTEM
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective:

- To understand the various types and functions of Operating Systems
- To learn the process and various scheduling algorithms
- To explore various issues in inter-process communications and address them
- To evaluate the Memory management algorithms, allocation methods and virtual memoryimplementations
- To analyze file and I/O management in Operating System
- To learn shell scripting fundamentals

Course Learning Outcomes: Completion of this course, students should be able to do the following:

- Understand fundamental operating system architectures, processes, threads, files, semaphores, Inter Process Communication, shared memory regions, etc.
- Analyze algorithms like Process scheduling and memory management algorithms.
- Categorize the operating system's resource management techniques, deadlock management techniques.
- Able to understand paging of memory, page replacement algorithms, file systems.
- Understand the working of secondary storage devices.

Unit-1: Overview of operating system and process management

- Introduction to Operating System: Basic Concepts and Terminology, Logical View, User View, Concept of Virtual Machine, Interrupt Concept. Role of Operating System Resource Manager in Memory Management, Device Management.
- Process Concepts, Process Control Block, Process Scheduling, Criteria, algorithms and Evaluation, Job Scheduling, Inter-Process Communication, Communication in Client-Server.

Unit-2: Process synchronization & deadlock and memory management

- Concept of Synchronization: Requirements, Mechanisms, Critical Section Problem and Monitors. Deadlock, prevention & avoidance, Deadlock Detection, Deadlock Recovery.
- Memory Management: Concept &Techniques, Contiguous & Non-Contiguous allocation, Logical & Physical Memory Conversion of Logical to Physical address, Paging, Segmentation, Segment with paging, Virtual Memory Concept, Demand paging: Page Replacement Algorithms, Allocation of Frames, Page fault.

Unit-3: File management and disk management

- File Structure: Protection, FILE system Implementation, Directory structure, Free Space Management, Allocation Methods, Efficiency & Performance, Recovery
- Disk Structure: Disk scheduling algoritlun, Disk management, Swap Space concept and Management, RAID structure, Disk performance issues.

Unit-4: Operating system types

- Distributed Operating System: Difference Between Distributed & Centralized OS, Advantages of Distributed OS, Types of Distributed OS, Concept of Global OS, NOS Architecture.
- Linux Operating System: Introduction, Terminal V/s File Manager, File Permissions in Linux/Unix, Linux Environment Variables, Communication in Linux, Managing Processes in Linux, Introduction to Shell Scripting, System calls, Fork-Jon, Shared memory.

SUGGESTED READING :

1. Andrew Tanenbaum (2015). Modern Operating Systems4th Edition, Pearson
2. Sdberschatz, Galvin, Gagne (2008). Operating System Concepts, 8th Edition, Wiley,
3. Dhananjay Dhamdherem (1999). Systems Programming & Operating Systems, 2nd Edition, Tata McGraw-Hill,
4. Daniel P. Bovet and Marco Cesati (2006). Understanding the LINUX KERNEL, 3rd Edition, O'Reilly.
5. Robert Love (2013). Linux System Programming: Talking Directly to the Kernel and C. Library, 2nd Edition, O'Reilly,

MCAADL14034: MATHEMATICAL FOUNDATION OF COMPUTER SCIENCE
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To acquaint the students with mathematical/logical fundamentals including numerical techniques,
- To understand probability and graph theory that serve as an essential tool for applications of computer and information sciences.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Develop mathematical thinking and problemsolvingskills associated with research and writing proof
- Get ideas about sets and perform operations and algebra on set. Determine different properties of relations, identify equivalence and partial order relations. Identify function and their properties. Get idea about Group Theory.
- Get exposure to a wide variety of mathematical concepts used in computer science discipline like probability.
- Analyze the given propositions and finding results using mathematical logic operators.
- Understand and apply the concepts of graphs and trees. Analyze whether given graphs are isomorphic and different algorithms to find the shortest path.

Unit 1: Set and logic

- Set, subset, power set, properties of set, operations on sets, products on set, representation of set in computer partitions.
- Prepositions, Logical Operations, Tautologies, Negation, Contradiction, Contraposition, Logical Implication, Logical Equivalence, Normal Forms, Theory of Inference and Deduction, Predicates and Quantifiers.

Unit 2: Relation, function, Algebra and Lattice

- Relation and Function: Binary relation, types of relation, equivalence relation and equivalence class, closure of relation, function, different types of functions, recurrence relation, First order linear recurrence relation, The second order linear homogeneous recurrence relation with constant coefficients.
- Algebraic Structures: Introduction to algebraic Structures and properties. Types of algebraic structures: Semi group, Monoid, Group, Abelian group and Properties of group. Subgroup.
- Lattices: Introduction, Partial ordered sets, Combination of Partial ordered sets, Hasse diagram, Introduction of lattices, Properties of lattices – Bounded, Complemented, Modular and Complete lattice.

Unit-3: probability and Statistics

- Basic probability- Addition and Multiplication theorem-problems, Conditional probability.
- Random variables – Discrete random variables, Joint distribution, Binomial and Poisson Distributions, Measures of central tendency- mean, median, mode for grouped data, Measures of dispersion, Mean deviation and Standard deviation,

Unit 4: Graph theory

- Basic concepts- finite and infinite graphs, incidence and degree, isolated and pendant vertices, null graph; Paths and Circuits- isomorphism, subgraphs, walks, connected and disconnected graphs and components, Euler graphs, Bi-partite graphs, Hamiltonian paths and circuits; Trees- properties of trees, rooted and binary trees, spanning trees, fundamental circuits, spanning trees in weighted graphs; Cut-sets- properties; Matrix representation of graphs: incidence matrix, sub matrices, circuit matrix, cut-set matrix, path matrix, adjacency matrix, Directed graphs.

SUGGESTED READING:

1. S Santha, Discrete mathematics, Cengage Learning
2. Veerarajan, Discrete Mathematics, TMGH
3. Liu and Mahapatra, Discrete Mathematics, TMH
4. Rosen : Discrete Mathematics, 6/e, TMH Kolman : Discrete Mathematics, PHI.
5. J.P. Trembely&R.Manohar, "Discrete Mathematical Structure with application to Computer Science", McGraw Hill.

MCAADL14044: DIGITAL SYSTEM AND COMPUTER ORGANIZATION
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To offer students basic and intermediate levels of knowledge on digital logic, assembly language, and computer organization.
- To enable students to design and evaluate logic circuits, pipelined processors, and cache memories.
-

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Will be able to understand the theory and architecture of central processing unit.
- Understand the functionality of central processing unit.
- Understand the I/O and memory organization.
- Explain the number systems, binary addition and subtraction, 2's complement representation and operations with this representation.
- Examine the structure of number systems and perform the conversion among different number systems.
- Illustrate reduction of logical expressions using boolean algebra, karnaugh-map and tabulation method and implement the functions using logic gates.
- Understand combinational circuits.
- Design and analyze synchronous and asynchronous sequential circuits using flip-flops. CO9: Implement combinational logic circuits using programmable logic devices.

Unit-1: Number Systems

- Binary, Octal, Hexa decimal numbers, base conversion, addition, subtraction of binary numbers, one's and two's complements, positive and negative numbers, character codes ASCII, EBCDIC etc.
- Computer Arithmetic: Addition and Subtraction, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations.
- Structure of Computers: Computer types, Functional units, Basic operational concepts, Von-Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer, Digital Logic Circuits: Logic gates, Boolean algebra, Map Simplification. Combinational Circuits: Half Adder, Full Adder, flip flops.
- Sequential circuits: Shift registers, Counters, Integrated Circuits, Mux, Demux, Encoder, Decoder. Data representation: Fixed and Floating point, Error detection and correction codes.

Unit-2: Basic Computer Organization and Design

- Instruction codes, Computer Registers, Computer Instructions and Instruction cycle. Timing and Control, Memory-Reference Instructions, Input-Output and interrupt. Central processing unit: Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC), CISC vs RISC

Unit-3: Register Transfer and Micro-operations

- Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logic shift unit. Micro-programmed Control: Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit. Input Output: I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, DMA. Instruction level parallelism: Instruction level parallelism (ILP)-over coming data hazards, limitations of ILP.

Unit-4: Memory System

- Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance considerations, Virtual memory, Paging, Secondary Storage, RAID.
- Multiprocessors And Thread level Parallelism: Characteristics of multiprocessors, Multi-Threaded Architecture, Distributed Memory MIMD Architectures, Architecture of Multi-Threaded Processor, principle of Multi-Threading, Interconnection structures, Inter Processor Arbitration, Inter processor Communication and Synchronization, Cache Coherence.

SUGGESTED READING:

1. Mano, M.M.: Digital Logic and Computer Design, Pearson.
2. Rajaraman V, Radhakrishnan : An introduction to Digital Computer Design. PHI
3. Mano, M.M.; Computer System Architecture, Pearson
4. Hamacher, Vranesic, Zaky: Computer organization, McGraw Hill.
5. Uyemura . J.P.: Digital Systems Design an integrated approach, Cengage
6. Leach : Digital Principles & Applications,7/e, TMH 7 Tocci : Digital Circuits, PE.

MCAADL14054: LAB-1: PROGRAMMING USING C AND DATA STUCTURE
(Credit 4, Total Marks=100)
(Final Practical = 70 Marks, Internal Practical = 30 Marks)

Course Learning Objectives:

- Be familiar with basic techniques of algorithm analysis
- Be familiar with writing recursive methods
- Master the implementation of linked data structures such as linked lists and binary trees
- Be familiar with advanced data structures such as balanced search trees, hash tables, priority queues and the disjoint set union/find data structure
- Be familiar with several sub-quadratic sorting algorithms including quicksort, merge sort and heapsort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree
- Master the standard data structure library of a major programming language
- Master analyzing problems and writing program solutions to problems using the above techniques.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Apply branching and looping for solving computational problem.
- Design and implement algorithms using arrays and strings.
- Solve problems using multi-dimensional arrays.
- Apply the concepts of functions and pointers to arrays and string
- Experiment with structure, Union files.
- Implement and manipulate fundamental data structures, Apply data structure operations effectively.
- Debug and troubleshoot data structure implementations, Analyze the efficiency of data structures. implement algorithms using data
- Design and structures, Experiment with advanced data structures and algorithms.

Unit-1: Operators and expressions and branching and looping

1. Program (and draw a flowchart) that reads three numbers, calculates and prints their sum and average.
2. Program that calculates and outputs the area, perimeter of a rectangle whose length is L cm, width is W cm.
3. Program to calculate the area of a circle where radius R is given.
4. Program to read the temperature in centigrade degree (C) and change it into Fahrenheit (F) and vice versa, where a.F = (9*C+160)/5 and b .C = (5*F-160)/9.
5. Program to read the coefficients a, b, c of a quadratic equation and find the roots of the equation.
6. Program to read two numbers that represent the length of two sides of a right triangle; compute and print the length of the hypotenuse (which is the square root of the sum of the squares of the lengths of the two sides).
7. Program to swap two numbers (Using and without using third variable).
8. Program to find the largest among three numbers.
9. Program to reverse a given number. Ex: 325 -> 523
10. Program to accept an integer and find the sum and product of all the digits.
    Ex: 325->3+2+5=10
11. Program to compute Power of XY

12. Program to generate all prime numbers within range 1-100.
13. Program to generate all odd/even numbers within range 1-100.
14. Program to generate Fibonacci Series up to N terms. [1,1,2,3,5,8,13…….]
15. Program to find GCD of two numbers.
16. Program to check whether a number is a Armstrong no. of the following form or not: 371->33+73+13.
17. Program to compute the summation of following series:
    a. 1+2+3+.....n
    b. $1+x+x^2+x^3+.....+x^n$
    c. $1+x+x^2/2! + x^3/3! .....+ x^n/n!$
    d. $1-x+x^2/2! - x^3/3! .....+ x^n/n!$

Unit-2: Array, string, Structure-union

1. Program to find Sum and Average of numbers in an array.
2. Program to find an element in an array using:
    a. Linear Search
    b. Binary Search.
3. Program to insert and delete elements in an array.
4. Program to sort a set of numbers using:
    a. Bubble Sort
    b. Selection Sort
    c. Insertion Sort.
5. Program to convert a decimal number into binary number and visa versa.
6. Program to add and subtract two Matrix.
7. Program to Multiply two Matrix.
8. Program to implement Stack data structure.
9. Program to check whether a square matrix is magic square or not.
10. Program to compute: D = (A*B)-(C+D) [A, B, C, D are matrices]
11. Program to check whether a string is Palindrome or not.
12. Program to find length of a string.
13. Program to implement string copy, compare, sub-string etc.
14. Program to manipulate employee records.
15. Program to add and subtract two complex numbers using structure.

Unit-3: Function and pointer, file handling

1. Program to find Factorial of a number using Recursive function.
2. Program to find GCD of two numbers using Recursive function.
3. Program to generate Fibonacci Series using Recursive function.
4. Program to implement above problem (1, 2, 3) using user-defined function.
5. Program to reverse an entire array using function.
6. Write C Program to demonstrate Call by value and Call by address.
7. Write C Program to find Sum and Average of numbers in an array by passing array to a function using Pointer.
8. Write C Program to find length of string using Pointer.
9. Write C Program to multiply two matrices using pointer.
10. Program to implement Link List using Self-referential structure.
11. Program to Open and Close a file.
12. Program to Copy one file to another file.
13. Program to merge two files.

Unit-1: Linear data structure ,non-linear data structure, Hashing and Graph

1. Program to implement Merge Sort and Quick Sort.
2. Program to implement Singly, Doubly, and Circular Linked list.
3. Program to implement Stack and Queue using array and linked list.
4. Programs to implement Infix to Postfix and Prefix conversion, Evaluation of expression.
5. Program to implement Insertion and Deletion in a Binary tree.
6. Program to implement Insertion and Deletion in a Binary Search tree (BST).
7. Program to implement Inorder, Preorder and Postorder traversal on BST.
8. Program to implement insertion and deletion in B tree.
9. Program to implement insertion and deletion in AVL tree.
10. Program to implement double hashing technique to map given key to the address space. Also write code for collision resolution (linear probing).
11. Program to implement Breadth First search and Depth first search using linked
12. Representation of graph, Program to create a minimum spanning tree using Kruskal's algorithm.

SUGGESTED READING:

1. Y. Kanetkar (2004). Let us C, BPB Publication.
2. E. Balaguruswami (2004). Programming in ANSI C 2nd Edition, Tata Megraw Hill. Y. Kanetkar (2017). Let us Python 2nd Ed., BPB Publication.
3. E. Balagurusamy (2017). Introduction to Problem Solving with Python, Tata Megraw Hill.
4. Dierbach, Charles (2012). Introduction to Computer Science using Python: A Computational Problem Solving Focus, Wiley Publishing. 2012..

SEMESTER-II

MCAADL15064: DATABASE MANAGEMENT SYSTEM

(Credit 4, Total Marks=100)

(Theory: 50 Marks, Practical: 20 Marks, Internal: 30 Marks)

Course Learning Outcomes:

- Understand the basic concepts and the applications of database systems.
- Master the basics of SQL and construct queries using SQL.
- Understand the relational database design principles.
- Familiar with the basic issues of transaction processing and concurrency control.
- Familiar with database storage structures and access techniques.
- Also get ideas about NOSQL.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand and evaluate the role of database management systems in information technology applications within organizations.
- Will be able to understand entities and their relations.
- Explain functional dependency, Functional Decomposition, Normalization techniques.
- Will be able to use Structured Query Language (SQL).
- Draw Entity-Relationship diagrams to represent database application scenarios.
- Acquire the full understanding of the database management system theory as a basis for designing and implementation database applications.

Unit-1: Introduction to database and relational data model

- Database System Concepts, Database Systema Environment, Instances and Schemas, Database Languages and Interfaces, Classification of DSMS
- Client/Server Architectures DBMS, Relations and Integrity Constrains, Relational Algebra and Calculus.
- Entity Relationship (3) Model, Mapping ER diagram to relations, Enhanced ER Model

Unit-2: Query language processing and normalization

- Features of SQL and PL/SQL, Various DDL and DML commands, PL/SQL Structures and advantages, Importance of Query Processing, Steps for Query Optimization and approaches to Query Optimization.
- Importance of Normalization, Functional Dependencies, First Normal Form (UNF), Second Normal Form (2NF), Third Normal Form (307) and Boyce Codd Normal Form (BCNF)

Unit-3: Transaction processing and introduction to big data

- Transaction states, Desirable properties of Transactions (ACID), Schedules and Recoverability, Lock based protocols, Transaction Processing.
- Concurrency Control Techniques, Database Recovery Techniques, Big Data Characteristics, Types of Big Data, Big Dats Architecture, Features

Unit-4: NOSQL and advanced databases

- SQL vs NoSQL, Advantages of NoSQL, Types of NoSQL database, when should NoSQL be used, CAP theorem in NoSQL, Querying and managing NoSQL.
- Temporal Database, Object-oriented Databases, Multimedia Database, Distributed Databases, Mobile Databases.

SUGGESTED READINGS:

1. Elmarsri and Navathe (2017). Fundamentals of Database system, 7/e, Pearson.
2. Silberschatz, Korth and Sudarshan (2013). Database System Concepts, 6/e, McGraw Hill.
3. CJ Dates (2002). An Introduction to Database Systems, 7/e, Pearson.
4. Bipin C Desai (2012). An Introduction to Database System, Rev. Ed., Galgotias.
5. Subhashini Chellappan & Seema Acharya (2019). Big Data and Analytics, 2/e, Wiley.
6. Vince Reynolds (2016). Big Data for Beginners, Create space Independent Pub.
7. Seema Acharya (2020). Demystifying NoSQL, Wiley.
8. Malhar Lathkar (2019). Python Data Persistence: With SQL and NOSQL Databases, BPB Publ.
9. P Raj, P., & Deka, G. C. (2018). A deep dive into NoSQL databases: the fise cases and applications. Academic Press.

MCAADL15064 LAB: DATABASE MANAGEMENT SYSTEM LAB

1. Viewing all databases, creating a database, Viewing all Tables in a Database,
2. Creating Tables (with and without Constraints), Inserting/Updating/Deleting/Displaying Records in / from a Table, Altering a Table, Dropping/Truncating/Renaming Tables, Backing up/restoring a Database.
3. Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause),
4. Queries involving-Date Functions, String Functions, Math Functions.
5. Jain Queries: Inner Join, Outer Join, Subqueries: With IN clause, With EXISTS clause. Creating Views (with and without check option). Dropping views. Selecting from a view.
6. PI/SQL program using Basic Loop/ While Loop/ FOR Loop to insert ten rows into a database table.
7. PL/SQL program to illustrate iterations.
8. PL/SQL program to illustrate various operators such as arithmetic/relational/ comparison/logical/string.
9. PL/SQL program to illustrate decision making.
10. PL/SQL program to illustrate CURSOR
11. PL/SQL program to illustrate TRIGGER.
12. Database Design and Implementation for few scenarios such as Banking System, Library Management System etc.
13. To practice and implement various operations using NoSQL database(s)(MongoDB/DynamoDB/CouchDB/Neo4j/InfluxDB).

SUGGESTED READING:

1. Vikram Vaswani (2017). MySQL: The Complete Reference, McGraw Hill (Indian Ed.).
2. Kevin Loney (2008). Oracle Database 11g The Complete Reference, Oracle Press(Indian Ed.).
3. Ivan Hayross (2010). SQL, PL/SQL the Programming Language of Oracle, 3/e. BPB Publications.
4. Urman, Hardman, Mclaughlin (2004). Oracle Database 10g PL/SQL Programming, Oracle Press.

MCAADL15073: COMPUTER NETWORKS AND CYBER SECURITY
(Credit 3, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Proficient at solving computer networking problems in the workplace.
- Pursue productive careers in computer networking or a related computing field.
- Engaged in continuing professional development or professional societies in computer networking or a related computing field.
- Standards set forth by professional societies of which they are members.
- To learn and understand cyber security fundamentals
- To familiarize the various cyber/ internet crimes and their causes
- To analyze data privacy for cyber security management
- To develop skills on securing the system using safety tools.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand basics of Computer Networks, reference models -OSI, TCP/IP.
- Describe, analysis and evaluate datalink, network, and transport layer protocols.
- Understand the basic terminologies related to cyber security and current cyber security threat landscape. They will also develop understanding about the Cyberwarfare and necessity to strengthen the cyber security of end user machine, critical IT and national critical infrastructure.
- Draw complete understanding of the cyberattacks that target computers, mobiles and persons. They will also develop understanding about the type and nature of cybercrimes and as to how report these crimes through the prescribed legal and Government channels.
- Understand the legal framework that exist in India for cybercrimes and penalties and punishments for such crimes, It will also expose students to limitations of existing IT Act,2000 legal framework that in followed in other countries and legal and ethical aspects related to new technologies.
- Understand the aspects related to personal data privacy and security. They will also get insight into the Data Protection Bill. 2019 and data privacy and security issues related to social media platforms.

Unit-1: Computer networks, physical layer and data link layer

- Introduction to Computer Networks: Topologies, Local Area Networks Metropolitan Area Networks, Wide area Networks, Reference Models: OSI and TCP/IP and Protocol Suite.
- Data Link Layer: Framing, Error Detection and correction, Stop-and-Wait ARQ Go-back-n.
- MAC layer: ALOHA, Slotted ALOHA, CSMA protocols, Protocols: 802.3, 802.4, 802.5, 802.11 b/g/n.

Unit-2: Network layer, transport layer and application layer

- Network Lager: Routing, optimality Principle, Shortest Path Routing Algorithm, Distance Vector Routing. Congestion Control Routing: General principle of Congestion control, leaky bucket algorithm, Token Bucket Algorithm.

- TCP/IP architecture, I, ARP, DHCP, Internet routing protsenis: OSPF, BGP Transport Service Protocols TCP, UDP.
- Application Layer: World Wide Web DNS, SMTP, FTP, HTTPS

Unit-3: Overview of cyber security and cyber crimes

- Cyber security increasing threat landscape, Cyber security technologies Cyberspace, attack, attack vector, attack surface, threat, vulnerability exploit, exploitation, hacker, non -state actors, Cyber terrorism, Protection of end user machine, Critical IT and National Critical infrastructure Critical, Cyber-warefare, Case Studies.
- Cybercrimes targeting Computer systems and Mobiles data diddling attacks, spyware, logic bombs, DoS, DDoS, APTs, virus, Trojans, ransomware, data breach., Online scams and frauds- email scams, Phishing, Vishing, Smishing, Online job fraud, Online sextortion, Debit/credit card fraud, Online payment fraud, Cyberbullying, website defacement, Cybersquatting, Pharming, Cyber espionage, Crypto jacking, Darknet illegal trades, drug trafficking, human trafficking., Social Media Scams & Frauds impersonation, identity theft, job scams, misinformation, fake news cybercrime against persons cyber grooming, child pornography, cyber stalking., Social Engineering attacks, Cyber Police stations, Crime reporting procedure, Case studies.

Unit-4: Cyber law, data privacy and data security

- Cybercrime and legal landscape around the world, IT Act, 2000 and its amendments. Limitations of IT Act, 2000. Cybercrime and punishments, Cyber Laws and Legal and ethical aspects related to new technologies- AI/ML, IoT, Blockchain, Darknet and social media, Cyber Laws of other countries, Case Studies.
- Data protection principles, Big data security issues and challenges, Data protection regulations of other countries-General Data Protection Regulations (GDPR), 2016 Personal Information Protection and Electronic Documents Act (PIPEDA)., Social media- data privacy and security issues.
- Cyber security Plan- cyber security policy, cyber crises management plan, Business continuity, Risk assessment, Types of security controls and their goals, Cyber security audit and compliance, National cyber security policy and strategy.

SUGGESTED READING:

1. Andrew 5 Tanenbaum and David J Wetherall (2010). Computer Networks, 5th Ed. Pearson.
2. Behrouz A. Forouzan (2007). Data Communications and Networking, 5th Ed McGraw Hill Education.
3. Sumit Belapure and Nina Godbole (2011). Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley Indas Pvt. Ltd.
4. Dorothy F. Denning (1999). Information Warfare and Security, Addison Wesley.
5. Henry A. Oliver (2015). Security in the Digital Age: Social Media Security Threats and Vulnerabilities, Create Space Independent Publishing Platform.
6. Natraj Venkataramanan and Ashwin Shriram (2016). Data Privacy Principles and Practice, CRC Press.

MCAADL15084: OBJECT ORIENTED PROGRAMMING IN C^{++} AND JAVA
(Credit 4, Total Marks=100)
(Theory = 50 Marks, Practical = 20 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Designed to develop understanding of fundamental concepts of object-oriented programming
- Illustrate the object oriented concepts and develop solutions using C++ and a little bit of JAVA language, their design principles and tools.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand Object-Oriented Programming and Basics of C++ and Java.
- Comprehend and Implement Inheritance and Polymorphism in C++ and Java.
- Understand and Implement Threads, Array & String Handling in Java.
- Handle Exceptions, Develop GUI Programs, and Implement Networking in Java.
- Understand and Use Constructors, Destructors, and Method Overloading in C++ and Java.
- Implement Advanced Features like Virtual Functions and Abstract Classes.
- Utilize Java Utilities Package and Implement Synchronization in Java Threads.

Unit-1: Object oriented programming, introduction to C++ and java, class and objects

- Basic concepts of OOP (Abstraction, Encapsulation, Inheritance, Polymorphism), procedural programming vs. OOP. Features, Lexical Issues, Data Types, Variables, Arrays, Operators, Control Statements
- Concept of class and objects, Constructors, Destructor, Method overloading. Static methods.

Unit-2: Inheritance, polymorphism

- Types of inheritance, Defining derived class, Abstract class, Access specifiers public, private and protected; public and private inheritance, accessing base class members, ambiguity in multiple inheritance in C++, virtual base classes.
- Interface in Java. Compile time polymorphism-operator overloading, function overloading. Run-time polymorphism Virtual function, and pure virtual function in C++.

Unit-3: Threads in java, array & string handling in java

- Creation, Synchronization, Runnable Interface, Deadlock, Suspending. Resuming and stopping threads, Multithreading.
- Operation on String, Mutable & Immutable String, Using Collection Bases Loop for String, Tokenizing a String, Creating Strings using String Buffer, Use of Array List& Vector.

Unit-4: Exception handling, gui programming in java, networking in java

- Exceptions & Errors, Types of Exception, Control Flow, Use of try, catch, finally, throw, throws.
- Working with Windows, Graphics, and Text, AWT Controls, Layout Managers, and Menus, Images, Java Utilities (java.util Package).
- Connecting and communicating using socket.

SUGGESTED READING:

1. Balaguruswamy, E. (2017). Object oriented programming with C++, 7th Ed., McGraw Hill.

2. Stroustrup, Bjarne (2002). The C++ Programming Language, 3rd Ed., Pearson.
3. Schildt, Herbert (2017). Java-The Complete Reference, 10th Ed., Oracle Press.
4. Balagurusamy, E. (2006). Programming with Java: A Primer, 3rd Ed., McGraw Hill.
5. Eckel, Bruce (2006). Thinking in Java, 4th Ed., Prentice Hall.
6. Liang, Y. Daniel (2018). Intro to Java Programming, 10th Ed., Pearson.

MCAADL15084 LAB: OBJECT ORIENTED PROGRAMMING IN C^{++} AND JAVA LAB

1. Define a class Vehicle and create five different types of vehicles such as Car, Bus, Truck, Jeep, Bike and use member function input and display the attribute values.
2. Initialize above objects using different type of constructors and print their values using a member function.
3. Define a class Matrix and perform addition, subtraction and multiplication between the matrix. Use constructor, destructor in your program.
4. Use operator overloading (python) to perform above matrix operations.
5. Define a class String and compare two strings by overloading (python) various comparison operators.
6. Define a basic two-dimensional Shape class from which objects such as rectangle, circle which can be derived. Let the user specify the position, size, of drawing 2-D object.
7. Write programs to demonstrate the use of Multiple, Multi-level and Hierarchical inheritance to create different type of vehicles.
8. Use method overriding in above programs.
9. Use method overloading to add two strings (concatenation), two matrices and two Numbers and display the results.
10. Use method overloading to calculate the length of a string, and an integer array.
11. Demonstrate the use of exception handling while implementing Stack class with push() and pop() operations.
12. Use multiple threading to calculate smallest number in a large array of integers.
13. Use each thread to calculate the local smallest number on sub-array and finally find the global smallest from all local smallest numbers.
14. Use in-built library to implement Array data structure.
15. Use in-built library to implement List data structure.
16. Create your own package to implement all String related operations. Use your package to perform string related operations on user given strings.

MCAADL15094: DATA MINING AND WAREHOUSING
(Credit 4, Total Marks=100)
(Theory = 50 Marks, Practical = 20 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To learn and understand data mining techniques and warehousing
- To generate association rules from real-life datasets
- To analyze different clustering methods and classification tools for identifying similar groups of datasets
- To gather knowledge on building data warehouse for mining applications

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand the functionality of various data mining and data warehousing component.
- Recognize the strengths and limitations of various data mining and data warehousing models.
- Explain the analyzing techniques of various data.
- Describe different methodologies used in data mining and data ware housing.
- Compare different approaches of data ware housing and data mining with various technologies.

Unit-1: Data Warehousing

- Introduction to data warehousing, its characteristics, data types, application and scopes of data warehouse. Data warehouse types.
- Design of data warehouse: Data warehouse models, schema and their comparison with respect to actual applications, Process flows within data warehouse. OLTP and OLAP, Load manager, warehouse manager and query managers. Aggregations, Metadata.

Unit-2: Introduction to data mining and clustering

- Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective. Data Mining Techniques: A Statistical Perspective on Data Mining, Similarity Measures, Data Pre-processing.
- Similarity and Distance Measures,Different types of data in clustering, Partitional Techniques, Hierarchical Techniques, Density based Techniques, Incremental Clustering, Clustering with Categorical Attributes, Bi-clustering.

Unit-3: Classification and Prediction

- Basic About Classification and Prediction, Distance-Based Algorithms, Decision Tree-Based Algorithms, Artificial Neural Networks, Support Vector Machine, Random Forest.

Unit-4: Association rules

- Introduction, APRIORI, FP-Growth, Parallel and Distributed Algorithms, Incremental Rules, Quantitative rules.

SUGGESTED READINGS:

1. M. H. Dunham (2006). Data Mining: Introductory and Advanced Topics Pearson.
2. J. Han and M. Kamber (2011). Data Mining: Concepts and Techniques 3rd Ed., Morgan Kaufmann.
3. AK Pujari (2016). Data Mining Techniques 4th Ed., University Press.
4. Pang-Ning Tan et al (2016). Introduction to Data Mining, Pearson Current Literatures.

MCAADL15094 Lab: DATA MINING AND WAREHOUSING LAB

(Learners may use programming languages like C/C++/Python)

1) Write a computer program to implement A priori algorithm using two different data structures for storing the item sets in the level wise generation of candidate sets.
2) Write a computer program to implement A priori algorithm using hashing to store the item sets in the level wise generation of candidate sets.
3) Write a computer program to implement the Pincer Search algorithm.
4) Write a computer program to implement the DIC (Dynamic Item set) algorithm.
5) Write a computer program to implement the Promoted Border Algorithm.
6) Write a computer program to implement the k-means algorithm and test it using the different similarity measures stated in the syllabus.
7) Write a computer program to implement the PAM algorithm.
8) Write a computer program to implement the CLARA algorithm.
9) Write a computer program to implement the CLARANS algorithm.
10) Write a computer program to implement the DBSCAN algorithm.
11) Write a computer program to implement the BIRCH algorithm.
12) Write a computer program to implement the CART algorithm.
13) Write a computer program to implement the ID3 algorithm.

MCAADL150104: FORMAL LANGUAGE AND AUTOMATA THEORY
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Acquire the full understanding of the Automata theory as a basis for all computer science language design.
- Classify the types of Automata's namely FA, PDA, TM based on the storage mechanism employed by each automaton. Further, carry out the conversion of NFA to DFA and as well perform minimization of Finite Automatons using the minimization strategies.
- Differentiate the different variants of Automata and its computational power to recognize languages.
- Analyze variants of grammars namely Regular, Context-Free, Context- Sensitive etc., and thereby generate different languages accepted by variants of automata using the given production rules of grammar.
- Verify the correctness of an argument, identify limitations of some computational models and determine problems namely Halting Problem, Post Correspondence Problem and Undecidable problems about Languages.
- Design Finite Automata, Push Down Automata, Turing Machine using the transition functions and graphs; further, design various parser (Top Down and Bottom Up) by constructing parsing table

Unit-1: Theory of computation and finite automata

- Mathematical Preliminaries, Formal Language, Automata, Types of automata, Application of Automata Theory, Chomsky Hierarchy of Languages and Grammar, Formal definition of Grammar.
- Finite Automata and its types, Non-Deterministic Finite Automata (NFA / NDFA) and its equivalent Deterministic Finite Automata (DFA), Minimization of FA.

Unit-2: Regular expression, grammar and languages

- Regular Set (RS), Regular Expression (RE), Regular Grammar (RG), Regular Language (RL), FA and its equivalence with RE, Context Free Grammar (CFG).
- Context Free Language (CFL), Simplification of CFG, Normal formals in regard to CFG (Chomsky and Greibach Normal Forms (CNF and GNF)), Context Sensitive Grammar (CSG), Context Sensitive Language (CSL), Unrestricted Grammar, Recursively Enumerable Language, Closure Properties of RL, CFL, CSL and Recursively Enumerable Language.

Unit-3: Parsing and push down automata

- Parsing and its types, Parser (LL, LR), Derivation Techniques, Representation of Derivation / Parse Tree, Ambiguity in CFG, Design of any one type of Parser (LL/LR)
- PDA and its types. Non-Deterministic Push Down Automata (NPDA), NPDA and its equivalence with CFG

Unit-4: Turing machine, computational complexity &undecidability

- TM and its types, Universal TM, Halting Problem, Post Correspondence Problem, Measuring and Classifying Complexity, Church Turing Thesis and Undecidable problems about Languages.

SUGGESTED READINGS:

1. Peter Linz (2016). An Introduction to Formal Languages and Automata, 6/e, Jones & Bartlett.
2. Hopcroft, Motwani and Ullman (2008). Introduction to Automata Theory, Languages and Computation, 3/e, Pearson.
3. K.L.P. Mishra & N Chandasekharan (2006). Theory of Computer Science: Automata, Languages and Computation, PHI.
4. C.K. Nagpal (2011). Formal Languages and Automata Theory, Oxford Higher Education.
5. ShyamalenduKandar (2013). Introduction to Automata Theory, Formal Languages and Computation, 1/e, Pearson.
6. Vivek Kulkarni (2013), Theory of Computation, Oxford Higher Education.

SEMESTER-III (For Option A and C)

MCAADL 25014: DISTRIBUTED SYSTEM

Credit 4, Total Marks=100)

(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives

- To learn the concepts, architectures and models used in distributed systems
- To illustrate the event ordering techniques and distributed file systems to solve problems in an efficient, reliable and scalable way.
- To understand different consistency models and analyze security in distributed systems.
- To analyze P2P computing and distributed shared memory

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Students will be able to recognize the concepts of distributed systems.
- Learn about file systems.
- Understand operating system architecture.
- Learn about Process concepts, states, synchronization.
- Understand about security techniques, cryptographic algorithms, and digital signatures.

Unit-1: Introduction to distributed system

- Introduction to Distributed systems, examples of distributed systems, challenges, architectural models, fundamental models, Introduction to inter-process communications- external data representation and marshalling, client server communication, group communication, Case study: IPC in UNIX.
- Communication between distributed objects , Remote procedure call, Events and notifications, Java RMI case Study , Introduction to DFS, File service architecture, Sun network file system, Introduction to Name Services, Name services and DNS, Directory and directory services

Unit-2: Distributed operating system support

- The operating system layer, Protection, Process and threads, Communication and invocation, Operating system architecture.
- Introduction to time and global states - Clocks, Events and Process states - Synchronizing physical clocks - Logical time and logical clocks - Global states, Distributed debugging, Distributed mutual exclusion.

Unit-3: Transaction and concurrency control, distributed transactions

- Transactions, Nested transaction, Locks, Optimistic concurrency control, Timestamp ordering, Comparison of methods for concurrency control, Introduction to distributed transactions, Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery

Unit-4: Security and replication

- Overview of security techniques, Cryptographic algorithms, Digital signatures, Cryptography pragmatics, Replication, System model and group communications, Fault tolerant services, Highly available services, Transactions with replicated data.

SUGGESTED READINGS:

1. George Coulouris, Jean Dollimore, Tim Kindberg "Distributed Systems Concepts and Design" Third Edition – 2002- Pearson Education Asia.
2. M. Van Steen, A.S. Tanenbaum, Distributed Systems, Third Edition, CreateSpace Independent Publishing Platform, 2017.
3. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2011.
4. Garg VK. Elements of distributed computing. John Wiley & Sons, 2002.

MCASPL25024A: MACHINE LEARNING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To understand the basic theory underlying machine learning.
- To be able to formulate machine learning problems corresponding to different applications.
- To understand a range of machine learning algorithms along with their strengths and weaknesses.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.
- To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand machine learning basics: Programming vs. machine learning, applications, types, validation, and evaluation metrics
- Learn feature selection: Importance, filter, wrapper, hybrid methods.
- Gain knowledge of regression: Linear, non-linear, logistic, Poisson regression.
- Explore support vector machines (SVM): SVM classifiers, kernels, R/Python implementation.
- Familiarize with Bayesian classification: Bayes theorem, classifier, naïveBayes.
- Acquire neural network skills: Perceptron, feed-forward/back-propagation, recurrent, deep networks, R/Python implementation.
- Learn clustering: Partitional, hierarchical, density-based, validation, R/Python implementation.

Unit-1: Introduction to ml and feature selection

- Traditional programming vs. Machine learning. Application, Types of Learning. Basic framework, Validation, Cross-validation, Confusion Matrix, Precision- Recall, Accuracy, ROC.
- Features and selection, Importance, Filter, Wrapper, Hybrid method.

Unit-2: Regression & support vector machines

- Linear Regression, Non-Liner Regression Model, Logistic Regression, Poisson Regression.
- SVM Classifier, Vector Kernels, Application-specific Kernels, Implementing SVM in R/Python.

Unit-3: Clustering and Classification

- Partitional Clustering, Hierarchical Clustering, Density based Clustering, Cluster validation measures, Implementing clustering in R/Python.
- Bayes Theorem, Bayes Classifier, Naive Bayes Classification.
- Neural Network: Single Layer Perceptron, Multilayer Perceptron, Feed-Forward and Back-Propagation, Recurrent Neural Network, Deep Neural Network, Implementing ANN in R/Python.

Unit-4: Advanced Learning

- Learning Sets of Rules – Sequential Covering Algorithm – Learning Rule Set – First Order Rules – Sets of First Order Rules – Induction on Inverted Deduction – Inverting Resolution – Analytical Learning – Perfect Domain Theories – Explanation Base Learning – FOCL

AlgorithmReinforcement Learning – Task – Q-Learning – Temporal Difference Learning. "Current Streams of Thought".

SUGGESTED READINGS:

1. Tom M. Mitchell (2017), Machine Learning, Me Grew Hills.
2. Ng. A. (2017). Machine learning yearning. (http://www.mlyearning. Org).
3. J Kalita, DK Bhattacharyya, S Roy (2023), Fundamentals of Data Science- Theory & Practice, Elsevier. S. J. Russell, P. Norvig. Artificial Intelligence: A Modern Approach. Third Edition, Prentice-Hall, 2010.
4. C. M. Bishop (2016), Pattern Recognition and Machine Learning, Springer.
5. Marco Gori , Machine Learning: A Constraint-Based Approach, Morgan Kaufmann. 2017
6. Ethem Alpaydin, Machine Learning: The New AI, MIT Press-2016

MCASPL25024B: BIOINFORMATICS
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Empower students with the skills to analyze biological data, develop computational tools, and solve complex biological problems using bioinformatics techniques.
- Equip students with skills to analyze biological data, develop computational models, and contribute to advancements in biomedical research and healthcare.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand Bioinformatics: Introduction, applications, DNA-RNA, proteins, databases, tools.
- Learn Sequence Analysis: DNA, protein sequences, alignment, scoring, multiple alignment.
- Explore Phylogenetic Analysis: Tree building methods, phylogenetic tree.
- Acquire Gene Identification skills: Prediction methods, homology-based, statistical, HMM.
- Learn Gene Expression Data Analysis: Microarray, RNA counts, clustering, expression profiles.
- Gain knowledge of Network Biology: Network models, gene regulatory network, protein interaction network, module detection.

Unit-1: Introduction to bioinformatics

- Introduction, Application, Central dogma of molecular biology, DNA-RNA, Amino acids, Protein, Major databases, biological data retrieval tools.

Unit-2: Sequence and phylogenetic analysis

- DNA and Protein Sequence, Next Generation Sequencing (NGS), FASTA format, Pair-wise sequence alignment, Dot Matrix methods, Scoring matrices, Dynamic Programming, Multiple sequence alignment, SP Method, Progressive Alignment Alignment-free sequence analysis.
- Phylogenetic Tree, Tree building methods (UPGMA, NJ maximum livelihood maximum parsimony).

Unit-3: Gene identification and prediction & gene expressiondata analysis

- Basis of Gene prediction, Gene Prediction Methods Homology based methods. Statistical and Hidden Markov Model.
- Microarray, RNA Read counts, Clustering &Bi-clustering of gene expression Profiles

Unit-4: Network biology

- Complex Network Models, Gene Regulatory Network-Inference & Analysis, Protein Interaction Network Inference & Analysis, Network module/complex detection.

SUGGESTED READINGS:

1. Bosu & S Kaur Thukral (2007), Bioinformatics: Experiments, Tools, Databases, and Algorithms, Oxford Higher Education.
2. Arthur Lesk (2014), Introduction to Bioinformatics, Oxford Higher Education.
3. PH Guzti& S Roy (2020), Biological Network Analysis: Trends, Approaches, Graph Theory, and Algorithms, Elsevier.
4. SC Rastogi et al. (2013), Bioinformatics: Methods and Applications, PHI.

MCASPL25024C: PATTERN RECOGNITION
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective:

- Introduce basic concepts and major techniques in statistical pattern recognition.
- These include concepts and techniques for data classification, feature selection, and dimensionality reduction.
- Introduce research development ability in pattern recognition through technical survey and presentation.
- Give ideas related to Fuzzy pattern recognition.

Course Learning Outcomes: Completion of this course, students should be able to do the following:

- Explain major techniques and algorithms in statistical pattern recognition and Fuzzy pattern recognition.
- Know basic concepts in other major approaches including syntactic methods.
- Get acquainted with recent developments in pattern recognition and its applications.

Unit-1: Bayes Decision Rules for two Class problem, Bayes maximum likelihood rule, minimum distance classifier, error probabilities for classifier, Mahala Nobis distance, Bound for error probabilities, Estimation of parameters, Learning.Single layer perceptron

Unit-2 : Clustering, Minimum within cluster distance critewrion, k-menas algorithm single linkage, complete linkage and average linkage algorithms, Isodata algorithm etc.

Unit-3 : Feature Selection

- Algorithms for feature selection such as Branch and Bound, Sequential forward and backward selections, GSFS and GSBS, (L, R) algorithm.
- Criterion function: Probabilistic Separability criterion, error probability based criterion, entropy based criterion, minimum within class distance based criterion, probabilistic independence.
- Principal Component Analysis

Unit-4 : Fuzzy Set-theoretic Pattern Recognition

- Usual Fuzzy set theoretic operations –union, intersection etc.
- Multivalued Logic: Zade Compositional Rule of inference
- Fuzzy *C-means* algorithm
- Supervised Classification: Multivalued Recognition System
- Fuzzy set theoretic based feature selection criteria

SUGGESTED READINGS:

1. Duda and Hart, “Pattern Classification ad Scene Analysis”, John Willey, 1990
2. P.A. Devijver and J. Kittler, “Pattern Recognition: A Statistical Approach”, 1983
3. K. Fukunga, “Introduction to Statistical Pattern Recognition”, Academic Press
4. S.K. Pal and Dutta Mazumdar, “Fuzzy Set Theroetic Methods for Patern Recognition”, John Willey, 1998.

MCASPL25034A: DATA SCIENCE
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective:

- To develop the ability for problem analysis and decision-making
- To gain practical, hands-on experience with statistics programming languages and big data tools through coursework and applied research experiences
- To apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively
- To develop the ability to build and assess data-based models

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand Data Science fundamentals: Concepts, principles, tasks, and applications.
- Learn Data Preparation: Types, normalization, missing values, feature selection.
- Explore Proximity Measures: Characteristics, types for numeric, categorical, and mixed data.
- Acquire Classification skills: Models, validation methods, metrics, Bayesian and Naive Bayes classification.
- Familiarize with Decision Trees: Rules, attribute selection, gain, Gini Index.
- Learn Distance-Based Classification: Nearest Neighbor Classifier.
- Gain knowledge of SVM and Neural Networks.

Unit-1: Introduction to data science and data preparation

- Concepts, Principle, Knowledge discovery and Data Science, what is not Data Science? Evolution of Data Science, Data Science Tasks, Applications.
- Data types-Nominal, Ordinal, Categorical, and Ratio, Normalization, Discretization, Missing Value Estimation, Sampling, Feature Selection.

Unit-2: Proximity measures and introduction to classification

- Proximity Measures and its Characteristics, Identity, Non-negativity, Symmetricity, Types of Proximity Measures-Numeric, Categorical, Mixed-type.
- Definition, Machine Learning Model, Classification and Prediction, Data Separability and Decision Boundary, Validation Methods, Cross-validation, Bootstrapping, Assessment Metrics Confusion Matrix, Sensitivity and Specificity, Accuracy.

Unit-3: Classification models

- Bayesian Classification, Naive Bayes Classification.
- Decision Tree: Decision Rules, Attribute Selection Measures, Gain, Gini Index, ID3
- Distance Based Classification: Simple Approach, Nearest Neighbour Classifier.
- Support Vector Machines: SVM Classifier, Vector Kernels, Application-specific Kernels, Implementing SVM using R/Python.
- Neural Network: Biological brain, Single Layer Perceptron, Multilayer Perceptron, Feed-Forward and Back-Propagation, Implementing ANN using R/Python.

Unit-4: Clustering and big data analysis

- Euclidean Distance, Manhattan Distance, Minkowski Distance, Partitional, Hierarchical, Density-based techniques, Assessment methods.
- Big Data, Overview, Big Data Analytics Architecture, Big Data Analytics Tool, Design of Big Data Mining Algorithms, Incremental and Distributed Algorithm.

SUGGESTED READINGS:

1. J. Kalita, DK Bhattacharyya, S Roy (2023), Fundamentals of Data Science Theory & Practice, Elsevier.
2. Tom M. Mitchell (2017), Machine Learning, Me Grew Hills.
3. Dunham, M. H. (2006). Data mining: Introductory and advanced topics. Pearson Education India.
4. Han, J., Pei, J., & Tong, H. (2022). Data mining concepts and techniques. Morgan kaufmann.
5. Vander Plas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc.

MCASPL25034B: CRYPTOGRAPHY AND NETWORK SECURITY
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Understand the fundamental principles of access control models and techniques, authentication and secure system design.
- Have a strong understanding of different cryptographic protocols and techniques and be able to use them.
- Apply methods for authentication, access control, intrusion detection and prevention.
- Identify and mitigate software security vulnerabilities in existing systems.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Explain the fundamentals of cryptography and understand classical encryption techniques. CLO2: Learn about how to maintain the Confidentiality, Integrity and Availability of data.
- Analyze and design block ciphers and stream ciphers.
- Understand hash functions, their types and threats.
- Analyze public-key cryptography, RSA and other public-key cryptosystems such as Diffie-Hellman Key Exchange.
- Analyze and design digital signatures.
- Understand and design user authentication techniques.
- Discuss public-key infrastructure (PKI), secure socket layer (SSL), and Kerberos.
- Learn System Modeling. Verification and performance analysis of security algorithms.

Unit-1: Introduction and stream ciphers

- Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective. Data Mining Techniques: A Statistical Perspective on Data Mining, Similarity Measures, Data Pre-processing.

Unit-2: Message digest, public-key parameters and intractable problems

- Properties of hash functions, MD2, MD5 and SHA-1, keyed hash functions, attacks on hash functions.
- Modular arithmetic, GCD, Primality testing, Chinese remainder theorem,modular square roots, finite fields.
- Integer, factorization problem, RSA problem, modular square root problem, discrete logarithm problem, Diffie-Hellman problem.

Unit-3: Public-key encryption and digital signatures

- RSA, Rabin and El Gamal schemes, side channel attacks. Key exchange: Diffie- Hellman and MOV.
- RSA, DSA and NR signature schemes, blind and undeniable signatures.

Unit-4: Entity authentication, network security and standards

- Passwords, challenge response algorithms, zero-knowledge protocols IEEE and ISO standards.
- Certification, public-key infra-structure (PKI), secure socket laver (SSL), Kerberon.
- Assignments System Modeling, assignment using Rhapsody system Verification, Assignment using SPIN, performance analysis assignment using Chronos

SUGGESTED READINGS:

1. Alfred J. Menezes et al. (1996). Handbook of Applied Cryptography, CRC Press.
2. William Stallings (2017). Cryptography and Network Security: Principles and Practice, 7th Ed., Pearson.
3. Jonathan Katz and Yehuda Lindell (2014). Introduction to Modern Cryptography, 2nd Ed., CRC Press, 2014.
4. Johannes Buchmann (2009). Introduction to Cryptography, 2nd Ed., Springer.
5. A Das, CAV Madhavan (2009). Public-Key Cryptography: Theory and Practice, Pearson.

MCASPL25034C: BIG DATA
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To gives an overview of Big Data, i.e. storage, retrieval and processing of big data.
- It also focuses on the "technologies", i.e., the tools/algorithms that are available for storage, processing of Big Data.
- It helps a student to perform a variety of "analytics" on different data sets and to arrive at positive conclusions

Course Learning Outcome:

Completion of this course, students should be able to do the following:

- Describe big data and use cases from selected business domains.
- Explain NoSQL big data management.
- Install, configure, and run Hadoop and HDFS.
- Perform map-reduce analytics using Hadoop.
- Use Hadoop related tools such as HBase, Cassandra, and Hive for big data Analytics, and understanding the recent trends in Big Data analytics

Unit-1: Understanding Big Data

What is big data, why big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data ,credit risk management, big data and algorithmic trading, big data and HealthCare, big data in medicine, advertising and big data, big data technologies, Introduction to Hadoop, open source technologies, cloud and big data mobile business intelligence, Crowd sourcing Analytics, inter and trans firewall analytics.

Unit-2: NoSQL Data Management

Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharing, masters slave replication, peer-peer replication, sharing and replication, consistency, relaxing consistency, version stamps, map reduce, partitioning and combining, composing map-reduce calculations.

Unit-3: Basics of Hadoop

Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop Distributed File System (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, oppression, serialization, Avro file-based data structures.

Unit-4: Map-reduce Applications

Map Reduce workflows, UNIT tests with MR UNIT, test data and local tests –anatomy of Map Reduce job run, classic Map-reduce, YARN, failures in classic Map-reduce and YARN, job scheduling, shuffle and sort, task execution, Map-reduce types, input formats, output formats.

SUGGESTED READINGS:

1. Michael Minelli, Michelle Chambers, and AmbigaDhiraj, "Big Data, Big Analytics: EmergingBusiness Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
2. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
3. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012.
4. Vignesh Prajapati, “Big Data Analytics with R and Hadoop”, 1 st Edition, Packet Publishing Limited, 2013.

MCASPL25044A: NATURAL LANGUAGE PROCESSING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Students will be able to comprehend the importance of using natural language processing when resolving issues in the real world.
- Enables students to apply and match the proper processing technique to a given situation.
- Students will be in a position to exhibit the necessary design abilities for large collection sets. Additionally, capable of understanding and presenting cutting-edge, sophisticated NLP research materials to an audience.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Discern the concept of POS tagging and CFG for the English language.
- Cognize the Vector Representation of words and skip-gram models
- Explore semantic analysis algorithms and deep learning techniques, to apply them in various NLP applications.
- Acquainted with Mathematical and programming tools for implementing NLP applications.

Unit-1: Basics of Machine Learning

Python Programming language, Basics of Probability, Introduction - terminologies - empirical rules – Statistical Properties of words – Probability and NLP – Vector Space Models - Pre-processing Tokenization, Parts-Of-Speech (POS) tagging, chunking, syntax parsing, Dependency parsing.

Unit-2: Vector Representation of words

Contextual Understanding of text – Cooccurrence of matrix – N-grams – Dense Word Vector. Word2Vec – CBOW and Skip-gram Models – One-word learning architecture- Forward pass for Word2Vec – Reduction of complexity – subsampling and negative sampling. Continuous Skip-Gram Model, GloVe, BERT, XLNet.

Unit-3: NLP Applications

Named Entity Recognition, Sentiment analysis, Text categorization using Machine learning algorithms, SVD and Latent semantic Indexing, Probabilistic Latent Semantic Indexing (pLSI) and Latent Dirichlet Allocation (LDA). Deep Learning for NLP: Neural Networks Basics, Feedforward Neural Network, Recurrent Neural Networks, LSTM, An Introduction to Transformers and Sequence-to-Sequence Learning.

Unit-4: Historical Approaches to Machine Translation

Statistical Machine Translation – Translation Models – Healthcare Data analysis and Text visualization: Summarizing lengthy blocks of narrative text, such as a clinical note or academic journal article. Answering unique free-text queries that require the synthesis of multiple data sources. Introduce Mathematical and programming tools to visualize a large collection of text documents.

SUGGESTED READINGS:

1. C.D. Manning et al, “Foundations of Statistical Natural Language Processing,” MIT Press. MIT Press, 1999. isbn: 9780262133609.
2. James Allen, “Natural Language Processing with Python”, O’Reilly Media, July 2009.

3. NiladriSekhar Dash and S. Arulmozi, Features of a Corpus. Singapore: Springer Singapore,2018, pp. 17– 34. ISBN: 978-981-10-7458-5.
4. Ian Goodfellow, YoshuaBengio, and Aaron Courville, Deep Learning, http://www.deeplearningbook.org. MIT Press, 2016.
5. NitinIndurkhya and Fred J Damerau, ”Handbook of natural language processing,” Chapman and Hall/CRC, 2010.
6. Daniel Jurafsky and James H. Martin ”Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,” 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000. isbn: 0130950696.

MCASPL25044B: INTERNET OF THINGS
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Describe what IoT is and how it works in current environment
- Recognize the factors that contributed to the emergence of IoT
- Design and program IoT devices
- Use real IoT protocols for communication

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand the fundamentals of the internet and Internet of Things (IoT), including transport services and socket programming.
- Apply network layer concept including forwarding, routing, IP addressing, DNS, and NAT, and understand mobile networking principles.
- Understand and implement real-time networking and become familiar with the definitions and architecture of lot.
- Identify the trend in IOT research, including data analytics, edge and fog computing, and IOT security.
- Understand and apply lot in various case studies, including home automation, health monitoring, smart agriculture, and smart cities.
- Differentiate between Machine-to-Machine (M2M) and IoT, and understand the management of lot, lot platforms, and languages,
- Gain proficiency in the use of lot for local area networks, including protocol such as Ethernet and Wi-Fi 802.11.

Unit-1: Internet and internet of things (IoT), transport services

- Layers, protocols, packets, services, performance parameters of a packet network as well as applications such as web, Peer-to-peer, sensor networks, and multimedia, TCP, UDP, socket programming.

Unit-2: Network layer, mobile networking

- Forwarding & routing algorithms (Link, DV), IP-addresses, DNS, NAT, and routers, Local Area Networks, MAC level, link protocols such as: point to pom! protocols, Ethernet, Wi-Fi 802.11, cellular internet access, and Machine to machine, Roaming and handoffs, mobile IP, and ad hoe and infrastructure less networks.

Unit-3: Public cloud environment

- Soft and real time, quality of service/information, resource reservation and scheduling, and performance measurements, Overview, applications, protocol & challenges, and architecture, Domains of loT, M2M via loT, Management of IoT, IoT platform, loT languages, IoT physical systems.

Unit-4: Betting up own cloud, future directions

- Data analytics using lot, Edge computing, Fog computing, Security in IOT. Home automation, Health monitoring, Smart agriculture and irrigation, Smart city and others.

SUGGESTED READINGS:

1. Chou, Timothy (2017). Precision: Principles, Practices and Solutions for the Internet of Things, McGraw Hill.
2. Hersent, Olivier et al. (2015). The Internet of Things: Key Applications and Protocols, Wiley.
3. MeEwen, A. &Cassimally, H. (2013). Designing the Internet of Things, Wiley.
4. Plister, (2013), Getting started with the Internet of Things, O'Reilly.
5. Hersent, O., Boswarthick, D., &Elloumi, O. (2015). The Internet of Things: Key Appstions and Protocols, Wiley.
6. Internet of Things: Architecture, Design Principles And Applications, Rajkamal, McGraw Hill HigherEducation.
7. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015.

MCASPL25044C: BLOCK CHAIN
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To assess blockchain applications in a structured manner.
- To impart knowledge in block chain techniques and able to present the concepts clearly and structured.
- To get familiarity with future currencies and to create own crypto token.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Describe the basic concepts and technology used for blockchain.
- Describe the primitives of distributed computing and cryptography related to blockchain.
- Illustrate the concepts of Bitcoin and their usage.
- Implement Ethereum block chain contract.
- Apply security features in blockchain technologies.
- Use smart contract in real world applications.

Unit-1: Introduction

- Need for Distributed Record Keeping, Modeling faults and adversaries, Byzantine Generals problem, Consensus algorithms and their scalability problems, Nakamoto's concept with Blockchain based cryptocurrency, Technologies Borrowed in Blockchain – hash pointers, consensus, byzantine fault-tolerant distributed computing, digital cash etc.

Unit-2: Basic Distributed Computing & Crypto primitives

- Atomic Broadcast, Consensus, Byzantine Models of fault tolerance, Hash functions, Puzzle friendly Hash, Collison resistant hash, digital signatures, public key crypto, verifiable random functions, Zero-knowledge systems

Unit-3: Bitcoin and Ethereum basics

- Bitcoin blockchain, Challenges and solutions, proof of work, Proof of stake, alternatives to Bitcoin consensus, Bitcoin scripting language and their use
- Ethereum and Smart Contracts, The Turing Completeness of Smart Contract Languages and verification challenges, Using smart contracts to enforce legal contracts, comparing Bitcoin scripting vs. Ethereum Smart Contracts, Writing smart contracts using Solidity & JavaScript.

Uinit-4: Privacy, Security issues in Blockchain

- Pseudo-anonymity vs. anonymity, Zcash and Zk-SNARKS for anonymity preservation, attacks on Blockchains: Sybil attacks, selfish mining, 51% attacks advent of algorand; Sharding based consensus algorithms to prevent these attacks.

SUGGESTED READINGS:

1. Narayanan, Bonneau, Felten, Miller and Goldfeder, "Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction", Princeton University Press.
2. Josh Thompson, 'Blockchain: The Blockchain for Beginnings, Guild to Blockchain Technology and Blockchain Programming', Create Space Independent Publishing Platform, 2017.
3. Imran Bashir, "Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained", Packt Publishing.

4. MerunasGrincalaitis, “Mastering Ethereum: Implement Advanced Blockchain Applications Using Ethereum-supported Tools, Services, and Protocols”, Packt Publishing.
5. Prof. Sandip Chakraborty, Dr. Praveen Jayachandran, “Blockchain Architecture Design And Use Cases”[MOOC], NPTEL: https://nptel.ac.in/courses/106/105/106105184.

MCASPL25054A: CLOUD COMPUTING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- Understand the fundamentals of Cloud Computing
- Learn about the various ways in which different cloud services can be applied to real world.

Course Learning Outcome: Completion of this course, students should be able to do the following:

- Understand and apply Virtualization Concepts, including various types of virtualization and high availability/disaster recovery strategies.
- Comprehend the building blocks of Cloud Computing and differentiate between public and private cloud environments.
- Utilize Amazon Web Services (AWS) for managing and creating virtual instances and storage volumes.
- Build and manage a private cloud environment using open-source tools and understand auto provisioning and integration of public and private clouds.
- Understand the use of cloud in lot and Big Data scenarios.
- Understand the domains and scope of work in cloud computing, including concepts like Platform as a Service (PaaS) and Software as a Service (SaaS)
- Gain insight into the market trends in cloud computing and its future.

Unit-1: Virtualization

- Virtualization Concepts, Types of Virtualization & its benefits, Introduction to Various Virtualization OS such Vmware, KVM etc. HA/DR using as Virtualization, Moving VMs, SAN backend concepts.

Unit-2: Cloud fundamentals, cloud as IAAS

- Cloud building blocks, Understanding public & private cloud environments. Private cloud environment such as: Basics of private cloud infrastructure, QRM cloud demo.

Unit-3: Public cloud environment

- Understanding & exploring Amazon Web services, Managing and Creating Amazon, EC2 instances, Managing and Creating Amazon EBS volumes, Tata Cloud details & demo, Managing Hybrid Cloud environment, Big Data, loT and Cloud.

Unit-4: Setting up own cloud, future directions

- How to build private cloud using open-source tools? Understanding various cloud plugins, setting up your own cloud environment, Auto provisioning, Custom images, integrating tools like Nagio, Integration of Public and Private cloud.
- Cloud Demain and scope of work, Cloud as PaaS, SaaS, Cloud Computing Programming, Introduction, Trends and market of cloud.

SUGGESTED READINGS:

1. Kavis, M. J. (2014). Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, & IaaS), Wiley.
2. Gonzales, D. (2010). Cloud Computing Bible: A Practical Approach To Cloud Computing Security, Cloud Problems To Be Aware of and More, Kindle E.
3. Erl, T. (2013). Cloud Computing: Concepts, Technology & Architecture, Prentice Hall.

4. Nielsen, L. (2014). The Little Book of Cloud Computing, Kindle E.
5. Bunker, G. & Thomson, D. (2006). Delivering Utility Computing, John Wiley & Sons.
6. Reese, G. (2009). Cloud Application Architectures, O'Reilly.
7. Gillam, L. (2010). Cloud Computing: Principles, Systems and Applications, Springer.

MCASPL25054B: SOFTWARE TESTING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To study the underlying concepts in software testing and to examine the various software testing issues
- Find their solutions Students are also made to expose to advanced software testing topics, such as object-oriented software testing activities, methods and tools.

Course Learning Outcomes:

- Identify the different software testing techniques, processes and errors handled in software projects
- Classify black box and white box testing techniques for functional and structural testing and test case designing
- Describe the different testing activities and levels of testing which aims to uncover the defects in all the stages of the project.
- Discuss the non-functional testing and debugging methods.
- Recognize the various issues for object-oriented testing and tools for testing

Unit-1: Introduction:

Introduction to software testing and analysis - Purpose of Software testing – Some Dichotomies – a model for testing - Error, Fault, Failure, Incident, Test Cases, Testing Process, Limitations of Testing - No absolute proof of correctness. Software testing Fundamentals - Specification-based testing techniques, code-based testing techniques, Model based testing.

Unit-2: Blackbox and Whitebox testing- Functional Testing:

Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique. Whitebox testing- Structural Testing: Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing, Static Analysis, Dynamic Analysis. Reducing the number of test cases: Prioritization guidelines, Priority category, Scheme, Risk Analysis, Regression Testing, Slice-based testing

Unit-3: Testing Activities and Object Oriented Testing:

- Unit Testing, Levels of Testing, Integration Testing, System Testing, Debugging, Domain Testing, Regression Testing, Acceptance testing,
- Issues in Object Oriented Testing, Class Testing, GUI Testing, Object Oriented Integration and System Testing, Methods of test data generation and validation.

Unit-4: Program slicing and its application, Reliability analysis, Formal methods; verification methods; oracles. Testing Tools: Static Testing Tools, Dynamic Testing Tools, and Characteristics of Modern Tools.

SUGGESTED READINGS:

1. William Perry, “Effective Methods for Software Testing”, John Wiley & Sons, New York, 2007.
2. Cem Kanerner, Jack Falk, Nguyen Quoc, “Testing Computer Software”, Second Edition, Van Nostrand Reinhold, New York, 2000.

3. Boris Beizer, “Software Testing Techniques”, Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
4. Louise Tamers, “Software Testing”, Pearson Education Asia, 2002
5. "Software Testing: A Craftsman's Approach, Second Edition," by Paul C Jorgensen, CRC Press, June 26, 2002. (required)
6. "The Art of Software Testing," 2nd ed., Glenford J. Myers, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004. (optional)
7. "Lessons Learned in Software Testing: a Context-Driven Approach," Cem Kaner, James Bach, and Bret Pettichord, John Wiley & Sons, Inc., New York, 2002. (optional).

MCASPL25054C: SYSTEM SOFTWARE
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objective

- Identify and describe the major components of system software, including the operating system, utility programs, and language translators.
- Explain the key functions of an operating system, including process management, memory management, file systems, and input/output operations.
- Understand the process of lexical analysis, syntax analysis, semantic analysis, optimization and code generation in compilers.
- Students will gain a comprehensive understanding of system software, its components, functions, and its critical role in the functioning of computer systems.

Course Learning Outcome

- These learning outcomes aim to provide a comprehensive understanding of system software, preparing students for advanced studies or careers in computer science and related fields.
- Understand Fundamental Concepts and components of system software, including operating systems, assemblers, linkers, loaders, and compilers.
- Understand the process of assembly language translation and the role of assemblers.
- Implement basic components of a compiler and understand the challenges involved incompiler design.
- Design and implement solutions to improve system software functionality and efficiency.

Unit-1: Overview:

Definition and classification of system software.

Unit-2: Assemblers:

Assembly language, Assembly process, Data structures, Macros andmacro processors.

Unit-3: Linkers and loaders:

Basic concepts, Static and dynamic linking, shared libraries,loaders, overlays. Case study of the Unix linking system, Windows DLLs, OLEs.

Unit-4: Compilers:

Introduction: Phases of a compiler, Languages and grammer, Chomshyhierarchy.Lexical analysis: Finite automata, Lexical analyzer, Lexical analyzergenerator (LEX). Parsing: Top-down and Bottom-up parsers, shift-reduce parser,recursive descent (operator precedence) parser, LL(1), LR parsers, Parser generator(YACC).

SUGGESTED READINGS:

1. Aho,A.V., Sethi, and Ullman J.d: complier design.
2. Dhandhere, System programming and operating systems, Tata McGrawHill.
3. Leland.L.Beck, System software, An introduction to System Programming, PearsonEducation
4. Louden, Compiler construction, Cengage

SEMESTER IV (For option A only)

MCASPL25064A: GRAPH THEORETIC ALGORITHM

Credit 4, Total Marks=100)

(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives:

- To teach learners how to create and design excellent algorithms.
- The course covers a wide range of important algorithms such as Depth First Search, Dijkstra's Shortest Path, Prim's Minimum Spanning Tree, and various Network Flow algorithms.
- See the applications of graphs in science, business and industry.

Course Learning Outcomes: Completion of this course, students should be able to do the following:

- Understand and explain the basic concepts of graph theory.
- Apply the basic concepts of mathematical logic.
- Analyze the basic concepts of mathematical logic.
- Evaluate some real time problems using concepts of graph theory.

Unit- 1: Introduction to Graphs

Overview of Graph Theory, Basic definitions: vertices, edges, directed vs. undirected graphs, Types of graphs: simple, weighted, directed, cyclic, acyclic

Unit-2: Graph Representation and Graph Traversal

Adjacency matrix, Adjacency list, Incidence matrix, Comparison of representations, Depth-First Search (DFS), Breadth-First Search (BFS), Applications of traversal algorithms

Unit-3: Shortest Path Algorithms and Minimum Spanning Trees

Dijkstra's algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm, Kruskal's algorithm, Prim's algorithm, Applications of minimum spanning trees

Unit-4: Advanced Topics and Review and Applications

Flow networks and the Max-Flow Min-Cut theorem, Ford-Fulkerson algorithm, Edmonds-Karp algorithmBipartite matching and assignment problem, Circulation and feasibility, Introduction to random graphs, Erdős–Rényi model, Basic properties of random graphs, Graph-based clustering algorithms, PageRank algorithm, Social network analysis, Introduction to spectral graph theory, Graph Laplacian and its properties, Applications in computer science

SUGGESTED READINGS:

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. "Algorithms" by Robert Sedgewick and Kevin Wayne
3. "Graph Theory" by Reinhard Diestel
4. N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI publication, 3rd edition, 2009.

MCASPL25064B: SPEECH PROCESSING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Learning Objectives: The goal of this course is to get the students excited about Speech Processing and to develop an understanding of:

- The acoustics of speech signals and corresponding articulatory details.
- Time and frequency based digital analysis of the speech signals.
- Machine Learning based robust, scalable, and adaptive speech processing.

Course Learning Outcomes: Completion of this course, students should be able to do the following:

- Understand the processes of human speech generation, transmission, and perception, and the mathematical models describing these physical processes.
- Develop a theoretical and practical (basic) understanding of the relevant branches of linguistics (articulatory and acoustic phonetics, and phonology), and signal processing (time and frequency-based analyses)
- Understand the Speech Processing pipeline from the design and collection of speech corpora, various feature extraction techniques, rule-based and Machine Learning based processing models, and appropriate evaluation techniques
- Develop a hands-on understanding of time and frequency-based speech processing techniques, Speech Recognition, and Speech Synthesis
- Gain hands-on experience with tools including Praat and audacity, languages including Matlab and Python, and toolkits including Sphinx and/or Kaldi.

Unit-1: Digital signal processing introduction and fundamentals of speech signal:

- signals, systems and signal Processing, Frequency in Continuous Time & Discrete Time Signals. Analog to Digital & Digital to Analog Conversion. Discrete Time Signals & Systems: Discrete Time Signals, Discrete Time Systems, Discrete Time Systems described by Difference equations, Correlation of Discrete Time Signals.
- History of speech recognition research, The Speech Signal : Speech production mechanism, Classification of speech, sounds, nature of speech signal, models of speech production. Speech signal processing: purpose of speech processing, digital models for speech signal, Digital processing of speech signals, Significance, short time analysis.

Unit-2: Time domain methods and frequency domain methods for speech processing:

- Time domain parameters of speech, methods for extracting the parameters, Zero crossings, Auto correlation function, pitch estimation.
- Short time Fourier analysis, filter bank analysis, spectrographic analysis, Formant extraction, pitch extraction, Analysis - synthesis systems. Homomorphic Signal Processing

Unit-3: Linear predictive coding of speech and speech analysis:

- Formulation of linear prediction problem in time domain, solution of normal equations, Interpretation of linear prediction in auto correlation and spectral domains.
- Cepstral analysis of speech, formant and pitch estimation, Mel frequency cepstrum computation, Applications of speech processing - Speech recognition, Speech synthesis and speaker verification.

Unit-4: Automatic speech recognition and Hidden Markov model for speech recognition:

- Basic pattern recognition approaches, Parametric representation of speech, Evaluating the similarity of speech patterns, Isolated digit Recognition System, Continuous digit Recognition System. Vector quantization, speech coding
- Introduction to Hidden Markov Model (HMM), Types of HMM, Hidden Markov Model (HMM) for speech recognition, Viterbi algorithm, Training and testing using HMMs, Adapting to variability in speech (DTW), Language models. Example of speech recognition project

SUGGESTED READINGS:

5. L. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1995, ISBN 0-13-015157-2
6. L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, 1978, ISBN 0-13-213603-1.

MCASPL25074A: MOBILE COMPUTING
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Leaning Objectives:

- Learning of the fundamental principles of mobile computing, the major technologies that support mobile computing, and a basic understanding of the role of mobile computing in the context of the everyday living.
- Gaining experience in implementing applications on the mobile device platforms.
- Gaining skills in solving technical challenges by working on labs.

Course Leaning Outcome:

- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions;
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline;
- Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline;
- Apply computer science theory and software development fundamentals to produce computing-based solutions;

Unit-1: Detailed Introduction of Mobile Computing:

History, Types, Benefits, Application, Evolution, Security Concern regarding Mobile Computing, Different Propagation Modes, Wireless Architecture and its types, needs of mobile user, Cellular system, Hexagonal geometry cell and concept of frequency reuse, Channel Assignment Strategies Distance to frequency reuse ratio

Unit-2: Telecommunication System and Mobile IP

- GSM: - Channel allocation, call routing Architecture, PLMN interface, addresses and identifiers, network aspects, frequency allocation, authentication and security, Handoffs Technique. GPRS: network operation, data services, Applications, Billing and charging
- Need of mobile IP, IP packet delivery, Agent Discovery, Registration, Tunnelling and encapsulation, Route optimization, IP Handoff

Unit-3: Mobile Transport Layer and Wireless Application Protocol:

- Overview of Traditional TCP and implications of mobility control. Improvement of TCP: Indirect TCP, Snoop TCP, Mobile TCP, Fast Retransmit/fast recovery, Time-out freezing, Selective retransmission, Transaction-oriented TCP.
- Introduction of WAP, WAP applications, WAP Architecture, WAP Protocol Stack, Challenges in WAP

Unit-4: Mobile Ad Hoc wireless networks and Introduction to 4G/5G:

- Introduction, Benefits, Difference, Routing protocols for ad hoc wireless networks: DSDV and AODV
- Introduction, features and challenges, Applications of 4G/5G, 4G/5G network architecture

SUGGESTED READINGS:

1. Schiller, Mobile Communications, Addison Wesley, 2003
2. Mehrotra, GSM System Engineering.
3. M V D Heijden, Understanding WAP, Artech House, 2000.
4. Charles Perkins, Mobile IP, Addison Wesley, 1999. 5. Charles Perkins, Ad hoc Networks, Addison Wesley, 2000

MCASPL250764B: ADVANCED DATABASE SYSTEM
(Credit 4, Total Marks=100)
(Theory = 70 Marks, Internal = 30 Marks)

Course Leaning Objectives:

- Strong foundation in advanced database concepts from an industry perspective.
- The database management system contributes with advanced data modelling concepts like OOD Modelling and ORD Modelling.
- The advanced database system arranges query processing and transaction management concepts for object-relational database and distributed database.

Course Leaning Outcomes: Completion of this course, students should be able to do the following:

- Design, develop and implement a mid-scale relational database for an application domain using a commercial-grade RDBMS
- Identify and resolve physical database design and implementation issues
- Use the persistence framework of a chosen language to perform Object Relational Mapping
- Research, analyse and use emerging technologies such as Big Data, NoSQL, On-Line Analytical Processing (OLAP) and Data Warehouses
- Have hands-on experience with a number of contemporary information management systems
- Explore a research aspect of advanced databases

Unit-1: Review of Relational Data Model and Relational Database Constraints:

Relational model concepts; Relational model constraints and relational database schemas; Update operations, anomalies, dealing with constraint violations, Types and violations. Object and Object-Relational Databases: Overview of Object Database Concepts, Object Database Extensions to SQL, The ODMG Object Model and the Object Definition Language ODL, Object Database Conceptual Design, The Object Query Language OQL, Overview of the C++ Language Binding in the ODMG Standard.

Unit-2: Disk Storage, Basic File Structures, Hashing, and Modern Storage Architectures:

Introduction, Secondary Storage Devices, Buffering of Blocks, Placing File Records on Disk Operations on Files, Files of Unordered Records (Heap Files), Files of Ordered Records (Sorted Files), Hashing Techniques, Other Primary File Organizations, Parallelizing Disk Access Using RAID Technology, Modern Storage Architectures. Distributed Database Concepts: Distributed Database Concepts, Data Fragmentation, Replication, and Allocation Techniques for Distributed

Unit-3: NOSQL Databases and Big Data Storage Systems:

Introduction to NOSQL Systems, The CAP Theorem, Document Based NOSQL Systems and MongoDB, NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j. Big Data Technologies Based on MapReduce and Hadoop: What Is Big Data? Introduction to MapReduce and Hadoop, Hadoop Distributed File System (HDFS), MapReduce: Additional Details Hadoop v2 alias YARN, General Discussion.

Unit-4: Enhanced Data Models:

- Introduction to Active, Temporal, Spatial, Multimedia, and Deductive Databases: Active Database Concepts and Triggers, Temporal Database Concepts, Spatial Database Concepts, Multimedia Database Concepts, Introduction to Deductive Databases. Introduction to Information Retrieval and

- Web Search: Information Retrieval (IR) Concepts, Retrieval Models, Types of Queries in IR Systems, Text pre-processing, Inverted Indexing, Evaluation Measures of Search relevance, web Search and Analysis. Trends in Information Retrieval

SUGGESTED READINGS:

1. Silberschatz and Korth, Database system concepts, McGraw Hill.
2. Elmasri and Navathe, Fundamentals of database systems; Narosa Publishing Co.
3. John G Hughes, Object Oriented Databases; Prentice Hall Int'nl Series in Computer Science
4. Andleigh and Thakrar, Multimedia Systems Design, Prentice Hall PTR
5. R Raghuramakrishnan& J Gehrke, Database Management System
6. Alhir, UML: In A Nutshell, O'Reilly

Web links and Video Lectures (e-Resources):

1. https://link.springer.com/book/10.1007/978-3-7091-2704-9
2. https://www.youtube.com/watch?v=ywTn9qHyI9I
3. https://www.youtube.com/watch?v=_qbKMdqQS6E
4. https://www.youtube.com/watch?v=PqPkYmRSQ_w
5. https://www.researchgate.net/publication/47393965_Data_warehousing_and_data_mining_A_case_study

MCASPL25084: SEMINAR
(Credit 4, Total Marks=100)
(Theory = 100 Marks)

Course learning Objectives:

- To create an environment to engage students in delivering and listening to interesting talks that promotes discussion
- To provide students with opportunity to learn new concepts and skills acquired in core courses and further extend these ideas to solve research/industry related problems
- Know how to read research papers critically and efficiently
- To learn fundamental principles, generalizations and important theories of Computer Science
- To enable students to find their own field of interest in academia, industry or entrepreneurship
- To help students develop their own learning and teaching styles and communication skills

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Explain factual knowledge (terminology, classifications, methods, trends). of current areas of research.
- State and explain some fundamental principles, generalizations, or theories the student has learned in this course.
- To apply gained knowledge in thinking, problem solving, or decisions making process.
- To achieve specific skills, competencies, and points of view needed by computing professionals.
- To judge the value of different contributions
- To identify promising new directions

Contents are:

A seminar should be given by an individual student, based on topics chosen from the emerging areas and technologies of Computer Science and Computer Applications. References from journals such as IEEE, ACM etc...shall be used. A report on this seminar with 15 – 20 pages should be prepared.

- Student shall present their work till literature survey.
- Student shall present their complete work.

SUGGESTED READINGS:

A list of works will be posted by mentors/teachers at the start of the course. The students also have the option of choosing works according to his/her own areas of interest.

MCADIS25098: PROJECT
(Credit 8, Total Marks=200)
(Practical = 200 Marks)

Course learning Objectives:

- Understand the process of Software development
- Solve real world computing problems with suitable software solutions

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Design solutions to real world computational problems.
- Select and use appropriate programming language/tools for solving problems.
- Perceive the art of verification and validation.
- Write technical reports.

The project (or dissertation) is one of the most vital subjects in the MCA program to be undertaken with the utmost seriousness in order to learn on tackling any challenging problem. The entire semester needs to be devoted to the dissertation work. The dissertation may involve software application development or novel problem-solving. It is recommended to work on any latest topic involving popular software tools and techniques. The student may perform his/her dissertation work within the department under the supervision of an supervisor.Student must submit a Detailed Project Report (60 to 80 Pages) in a format as specified by the department.

SUGGESTED READINGS:

Relevant literature and software tools for the chosen problem.

SEMESTER III(For Option B only)
MCADIS250120: RESEARCH THESIS/ PROJECT 1
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Understand the process of Software development
- Solve real world computing problems with suitable software solutions

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Design solutions to real world computational problems.
- Select and use appropriate programming language/tools for solving problems.
- Perceive the art of verification and validation.
- Write technical reports.

The Research thesis/Project is one of the most vital subjects in the MCA program to be undertaken with the utmost seriousness in order to learn on tackling any challenging problem. The entire semester needs to be devoted to the dissertation work. The dissertation may involve software application development or novel problem-solving. It is recommended to work on any latest topic involving popular software tools and techniques. The student may perform his/her dissertation work within the department under the supervision of an supervisor.Student must submit a Detailed Project Report (60 to 80 Pages) in a format as specified by the department.Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

Relevant literature and software tools for the chosen problem

SEMESTER IV (For Option B only)
MCADIS250220: RESEARCH THESIS/ PROJECT 2
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Understand the process of Software development
- Solve real world computing problems with suitable software solutions

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Design solutions to real world computational problems.
- Select and use appropriate programming language/tools for solving problems.
- Perceive the art of verification and validation.
- Write technical reports.

The Research thesis/Project is one of the most vital subjects in the MCA program to be undertaken with the utmost seriousness in order to learn on tackling any challenging problem. The entire semester needs to be devoted to the dissertation work. The dissertation may involve software application development or novel problem-solving. It is recommended to work on any latest topic involving popular software tools and techniques. The student may perform his/her dissertation work within the department under the supervision of an supervisor.Student must submit a Detailed Project Report (60 to 80 Pages) in a format as specified by the department. Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

Note: Research thesis/project with minimum 2 conferences papers. In each III and IV semester students have at last one conference paper. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

Relevant literature and software tools for the chosen problem

SEMESTER IV (For Option C only)
MCADIS250620: RESEARCH THESIS/ PROJECT
(Credit 20, Total Marks=500)
(Practical = 500 Marks)

Course learning Objectives:

- Understand the process of Software development
- Solve real world computing problems with suitable software solutions

Course learning Outcomes: Completion of this course, students should be able to do the following:

- Design solutions to real world computational problems.
- Select and use appropriate programming language/tools for solving problems.
- Perceive the art of verification and validation.
- Write technical reports.

The Research thesis/Project is one of the most vital subjects in the MCA program to be undertaken with the utmost seriousness in order to learn on tackling any challenging problem. The entire semester needs to be devoted to the dissertation work. The dissertation may involve software application development or novel problem-solving. It is recommended to work on any latest topic involving popular software tools and techniques. The student may perform his/her dissertation work within the department under the supervision of an supervisor. Student must submit a Detailed Project Report (60 to 80 Pages) in a format as specified by the department. Research thesis/project with minimum 1 conferences papers. Peer reviewed research publication should be encouraged.

SUGGESTED READINGS:

Relevant literature and software tools for the chosen problem